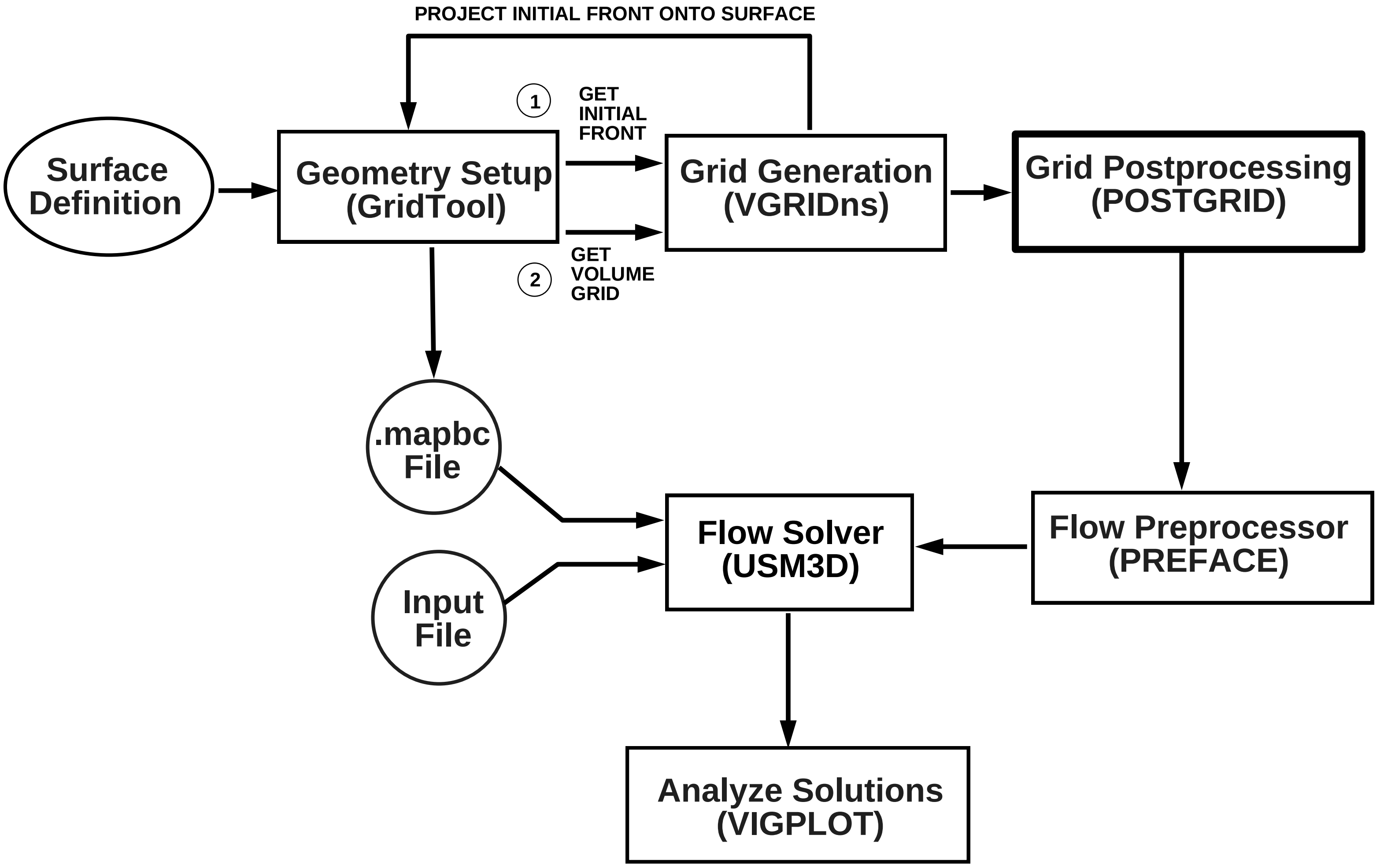
VGRID 3.2
Reference Documents

Written and Compiled by

Javier A. Garriz
ViGYAN, Inc.

Preliminary Draft
September 1998

Flowchart for TetrUSS



PROJECT INITIAL FRONT ONTO SURFACE

Surface Definition

Geometry Setup (GridTool)

1 GET INITIAL FRONT

2 GET VOLUME GRID

Grid Generation (VGRIDns)

Grid Postprocessing (POSTGRID)

.mapbc File

Input File

Flow Solver (USM3D)

Flow Preprocessor (PREFACE)

Analyze Solutions (VIGPLOT)

So you want to run VGRID 3.2?

Here are a few simple guidelines to remember as you construct your patches and distribute the background grid sources

What You Should Remember When Constructing Patches:

1. What will VGRID do with the patches I've constructed?

VGRID will perform a cubic spline fit of every curve comprising every edge of every side of every patch. This is done in order to obtain a continuous representation of the patch edge, along which VGRID will distribute grid points in a manner consistent with the user−specified spacing. As such, you should construct your curves in ways that prevent the usual difficulties associated with any spline−fitting procedure. Specifically:

a. Avoid building curves that contain sharp slope discontinuities. Specifying a patch edge with sharp slope breaks will cause the spline to "ring" and introduce curvature that is not actually present in the surface geometry. Keep in mind that while most of your patches will be 3 or 4−sided, each of the sides can be composed of multiple edges. VGRID will spline fit each of the edges individually. Therefore, the best way to preserve a slope break is to break the discontinuity into several edges (beginning and ending at the points of discontinuity), guaranteeing that (regardless of the user−specified grid size in the area) VGRID will place a grid point at the endpoints of each individual edge. Refer to the illustrations on the "What will VGRID do with the curves I've constructed?" pages that follow for a schematic of this case.

b. Avoid building curves that contain points distributed in a highly non−uniform way (i.e., having several points spaced very far apart next to several spaced very close together). This can also lead to "ringing" of the spline.

2. Why does patch shape matter?

Remember, the information contained in the .d3m file is the only input that VGRID receives and uses to generate the mesh. Therefore, regardless of whether your surface definition was composed of B−spline surfaces and/or curves, or simply discrete points, VGRID will use only the curves you created (and specified in the .d3m file as discrete points) to obtain the surface mesh. The .d3m file contains absolutely NO information pertaining to the shape of the surface(s) underlying each of the patches you created.

Once VGRID has spline−fitted each edge of each side of a patch, it uses the edge information to obtain the surface mesh on that patch. The shape of the resulting surface mesh on the patch, as you might expect, will be a blend of the shapes of the sides comprising the patch.

What You Should Remember When Constructing Patches, continued:

Hence, the more "well−behaved" the patch shape (i.e., the more closely the underlying surface is approximated by and reflected in your patch boundaries), the closer the surface triangles will be to the underlying surface(s). While it is true that you will (eventually) be projecting the generated surface triangles onto the appropriate underlying surfaces, it is still good practice to construct your surface patches in such a way as to minimize the amount of projection necessary to place each node of each surface triangle on the appropriate place on the actual surface definition. That way, any cell skewness that is present in the unprojected mesh is not worsened as a result of projection. Also (and very importantly), in areas of the geometry where tight, concave corners are present, badly−formed patches can lead to surface triangulations which deviate significantly from the actual surface (before projection). That is, a concave corner with an already small included angle might be rendered even smaller by a bad surface triangulation, increasing the chances that the grid generator will encounter difficulty (especially when generating viscous meshes).

Minimizing the amount of projection necessary also minimizes the chances of the resulting volume grid encountering problems introduced as a result of surface points being projected onto the wrong places (leading to the formation of "folded" surface triangles).

By far, the best way to achieve the greatest amount of control over the quality of the resulting surface triangulations is to follow the guidelines outlined in the page entitled "Some Guidelines for Constructing Patches for VGRID." Several examples of "good" and "bad" patches are shown on the page immediately following the "Guidelines" page. As you will notice, when constructing 4−sided patches, the "minimum" requirement for a "good" patch is that there be at least two sides that are approximately parallel to each other. In general, you will have the greatest amount of control if you use mainly (well−behaved) 4−sided patches, while limiting the number and spatial extent of 3−sided patches.

The time spent constructing good patches, while it may at first seem considerable, always pays dividends later in the grid generation process.

What You Should Remember When Distributing Background Grid Sources :

1. How do I know whether to place a source on the surface, or offset it some distance within the surface?

The simple guideline to remember is that if you want very localized clustering (such as, say, over the very leading edge of a wing), then place a source right on the surface. It’s effect should show up on the surface mesh as a tight circular (for nodal sources) or straight−line (for linear sources) cluster of cells.

What You Should Remember When Distributing Background Grid Sources, continued :

When a more diffuse, uniform distribution over a wider area is desired (as in, say, the mid−chord area of a wing or airfoil), place the sources just inside (or outside, depending on if you also want off−surface clustering) the surface of the configuration.

2. How do I know what values of a_n, b_n, etc. to assign to each source?

The magnitudes of the values you assign to the a_n and b_n source intensity parameters are important only relative to each other. That is, there is no "acceptable" or "unacceptable" range of values. The simple rule is to assign larger values to sources whose influence you want to be "felt" over larger regions of the mesh, and smaller values to sources whose influence you wish to keep relatively localized. Sources that you choose to locate right on the surface of the geometry will, therefore, tend to be assigned smaller values of the intensity parameters, while those "offset" from the surface (and whose role it is to provide the spacing value over a larger region of space) are given correspondingly larger values.

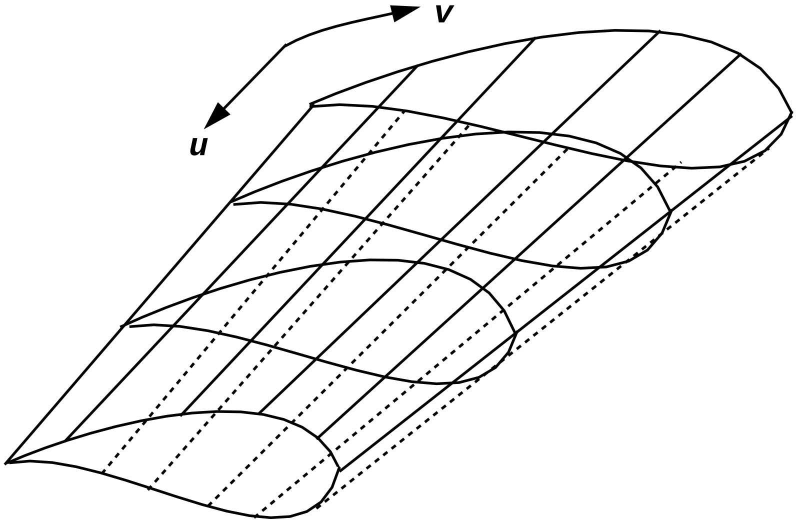
Another useful guideline to remember when assigning intensity values to linear sources is to scale the values proportionately according to the length of the linear source. In general, the effective "region of influence" of a given linear source (with a given a_n value) will decrease as it’s length increases. Hence, in order to exert influence over the same "amount" of space, the a_n value assigned to a linear source of a given length ought to be greater than the value assigned to a shorter linear source (intended to "cover the same amount" of space).

Keep in mind that you can exert a great deal of control over the resulting grid point distribution on the surface (and in the field) by simply assigning a_n values consistent with the guidelines given above, and not even using the additional directional control afforded you by the b_n parameter. In fact, it is recommended that during your first few attempts at generating grids, you only vary the a_n values, and not bother with varying the b_n parameter.

A Few Convenient Hot Keys to Remember When Running VGRID :

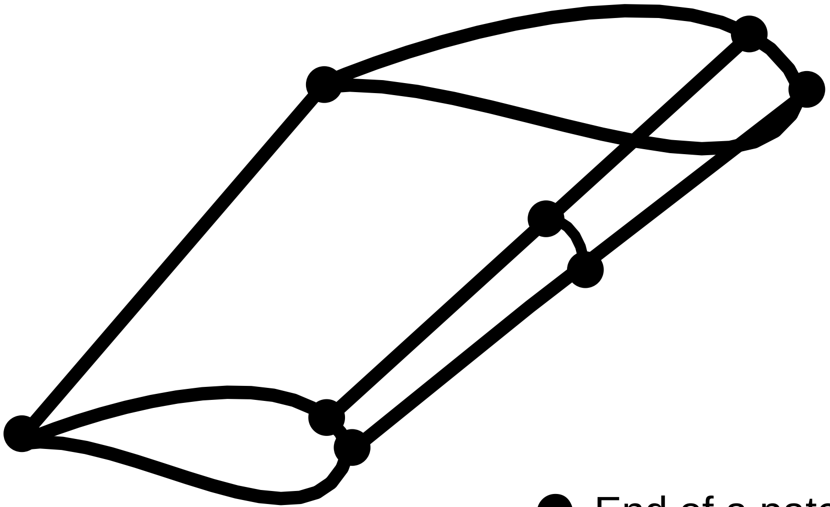
"ESC" key :	"Go on to the next menu"
Right−Mouse Click:	Used for registering your choice of a given menu item
"H" key :	Brings up the list of Hot Keys (including the function of each of the mouse keys for manipulating the geometry)

Your Mission....



To go from a surface definition...

...To a set of surface "patches"...

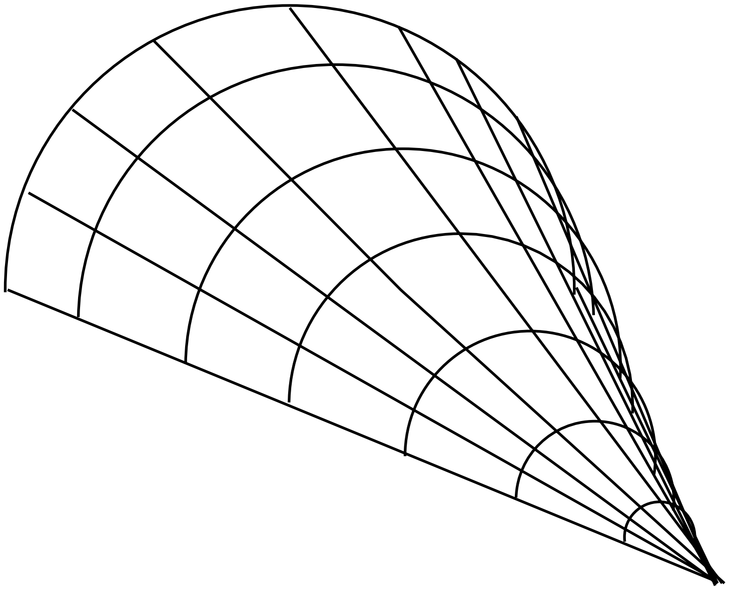


● End of a patch side

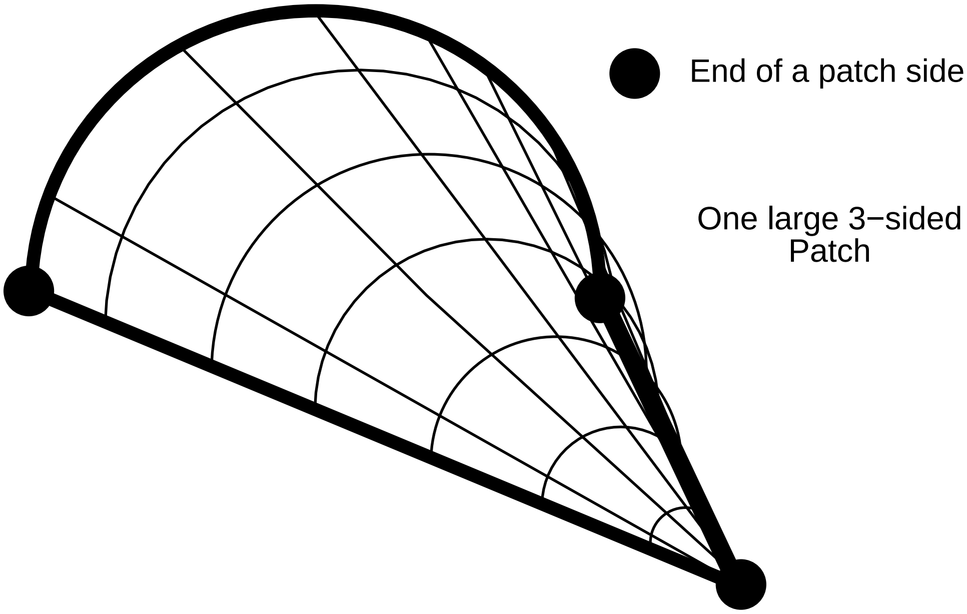
Patching Choices :

Remember, the shape of the surface triangulation on any patch will be a blend of the shapes of the patch sides

Given a surface like this ....



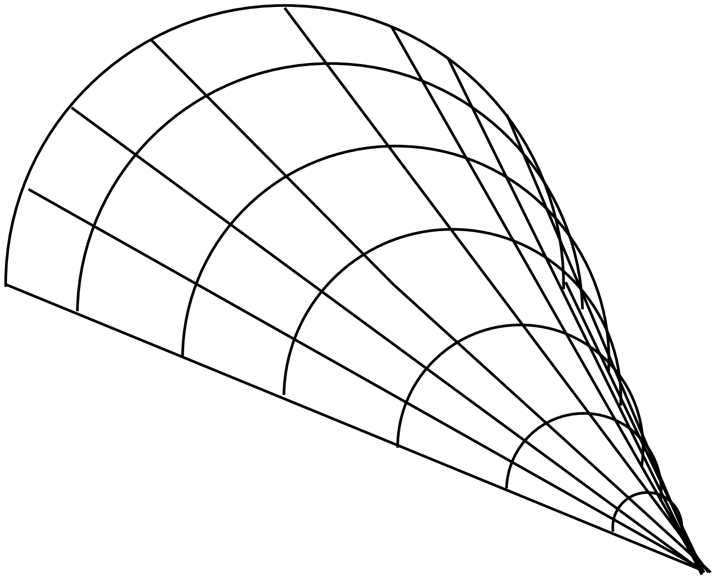
Could choose to patch it like this ....



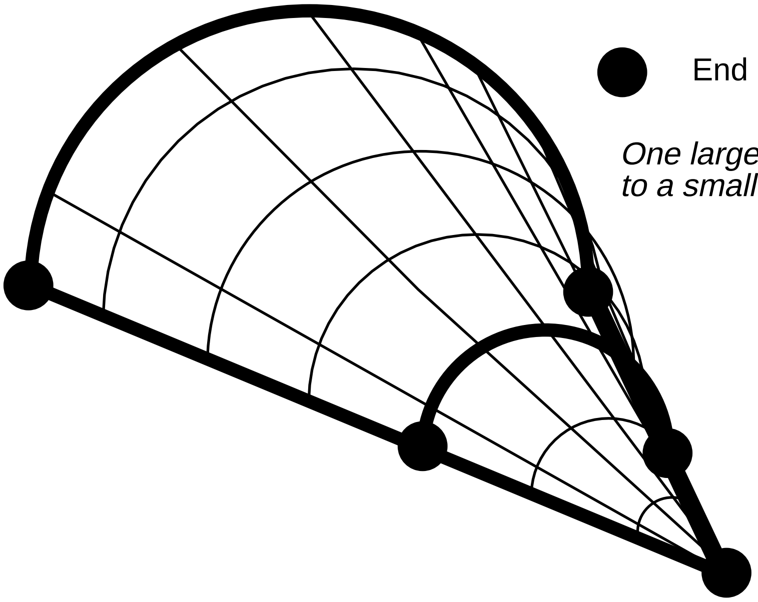
Patching Choices, continued :

OR

Given a surface like this ....



Could choose to patch it like this ....
(MUCH better)

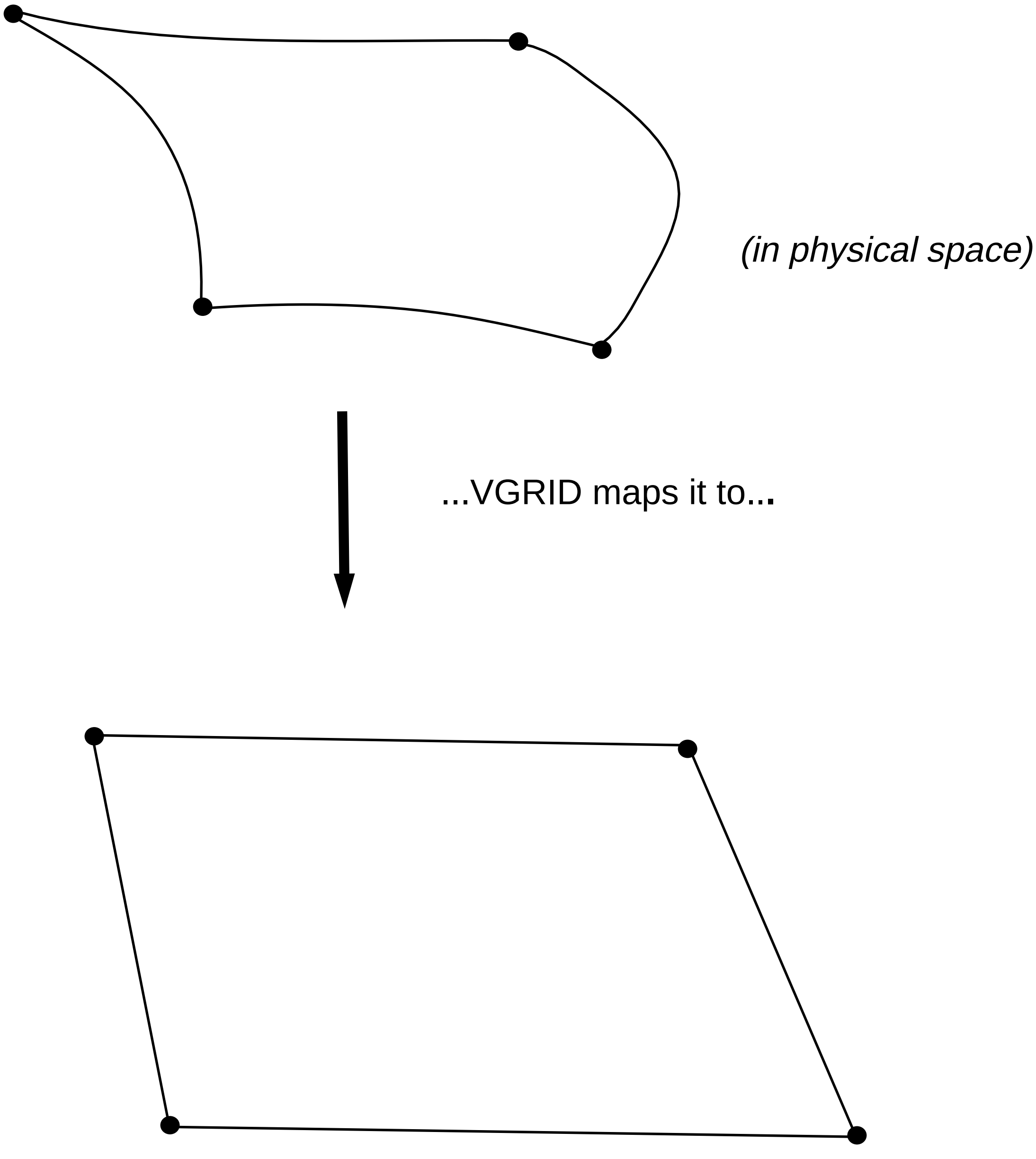


● End of a patch side

One large 4−sided patch next
to a small 3−sided patch

What does VGRID do with a patch?

Given a (non−planar, 4−sided) patch shaped like this...



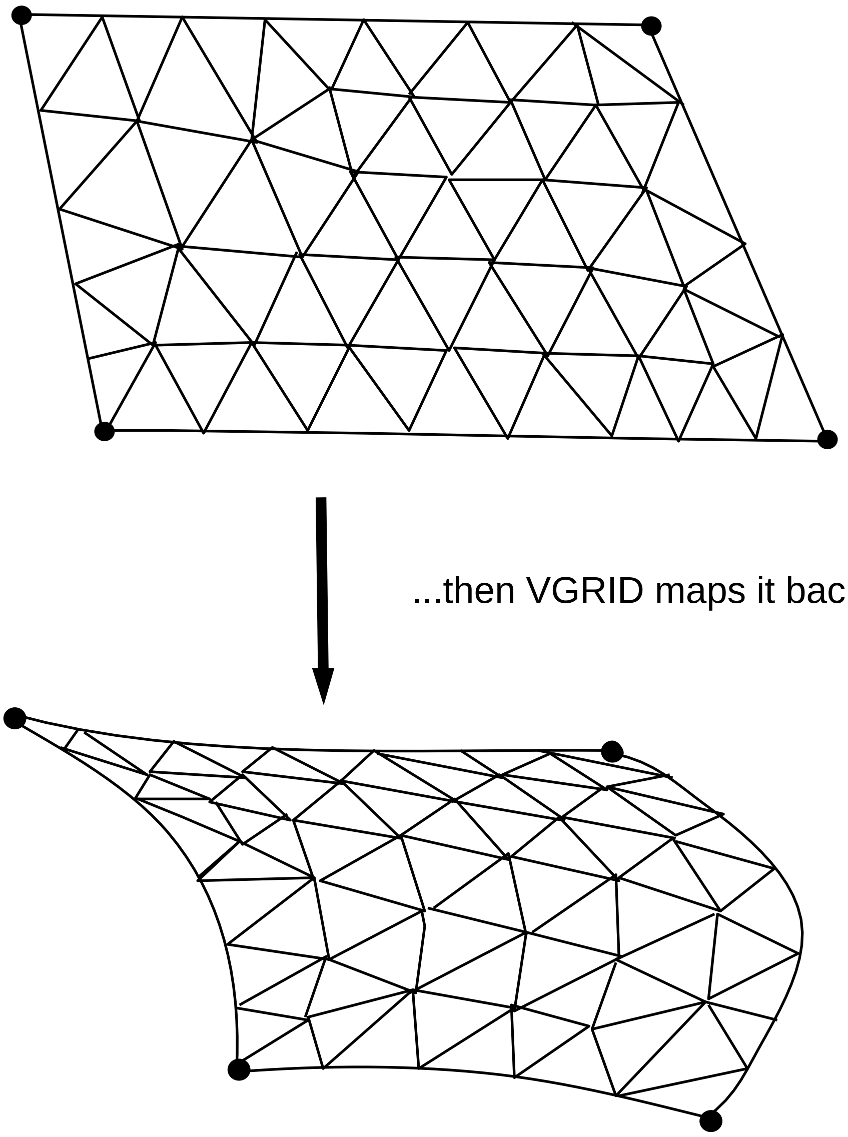
(in physical space)

...VGRID maps it to**...**

(in computational space)

What Does VGRID do with a patch? (continued)

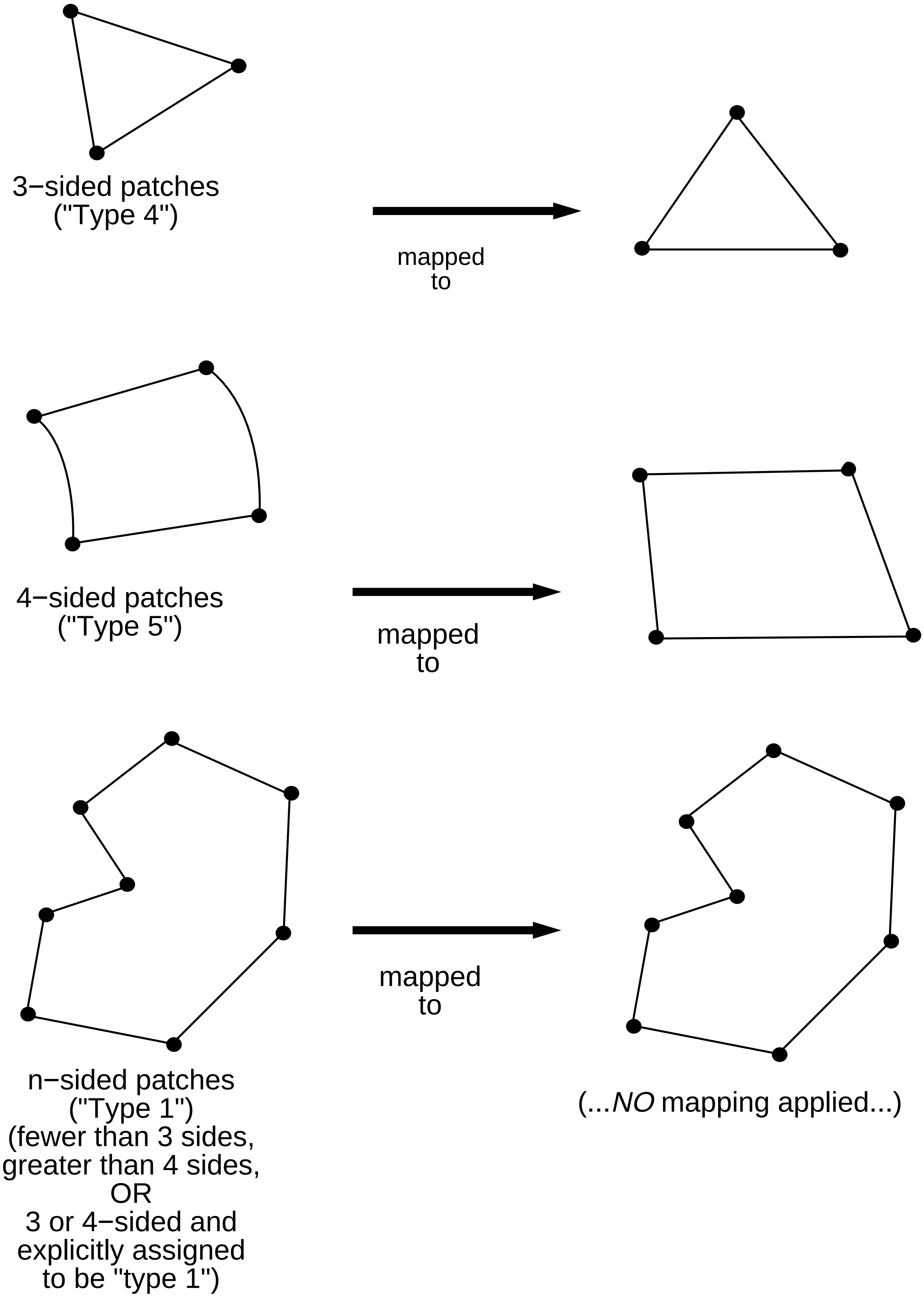
VGRID generates the surface mesh
in computational space....



...then VGRID maps it back to*...*

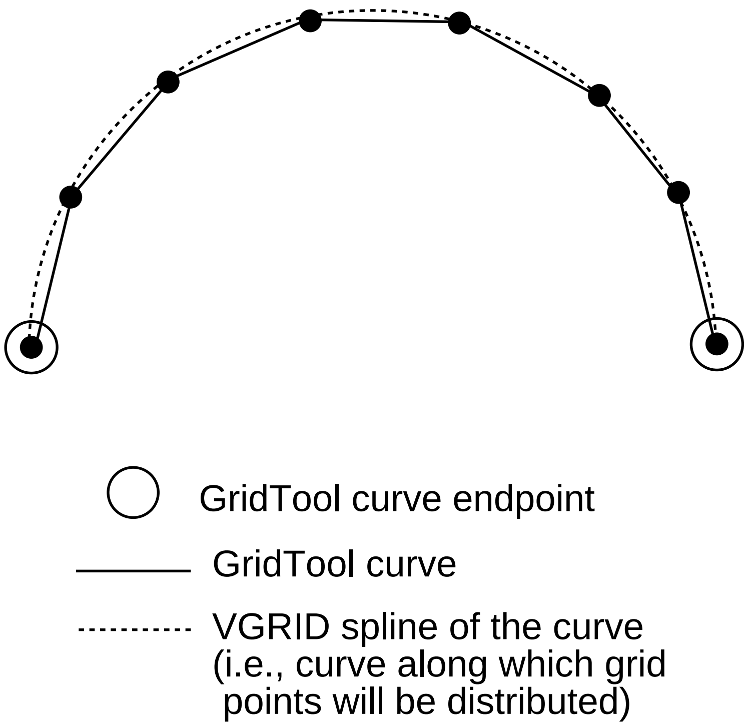
(...physical space)

NOTE: The number of triangles in the computational and the physical patch representations remains the same, obviously, even though that is not depicted in the schematic shown above.

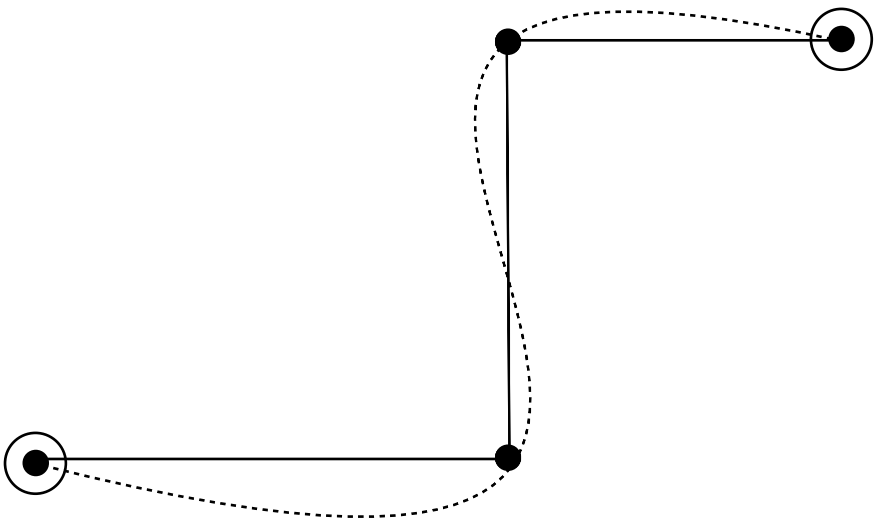

3−sided patches
("Type 4")
mapped
to
4−sided patches
("Type 5")
mapped
to
mapped
to
n−sided patches
("Type 1")
(fewer than 3 sides,
greater than 4 sides,
OR
3 or 4−sided and
explicitly assigned
to be "type 1")
(...*NO* mapping applied...)

What does VGRID do with the curves I've constructed?

(i.e., the curves that form each edge of each side
of each patch)

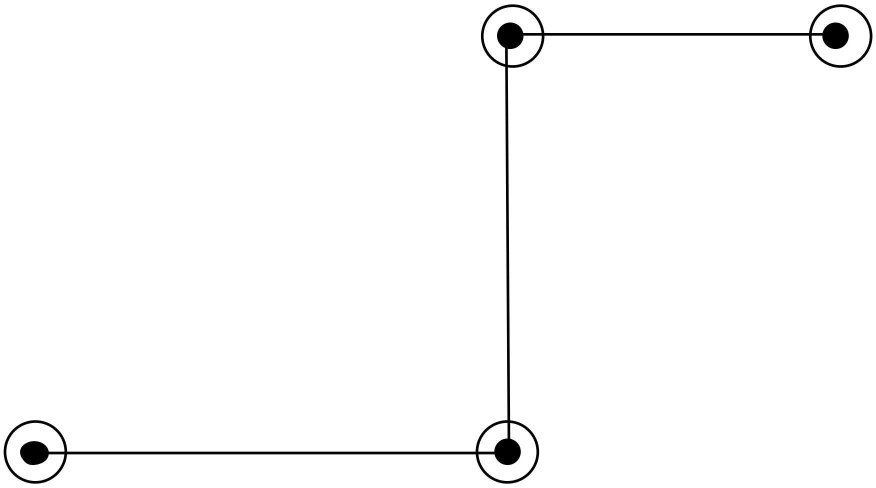

○ GridTool curve endpoint
—— GridTool curve
------ VGRID spline of the curve (i.e., curve along which grid points will be distributed)

One scenario to avoid :



In addition to the fact that non−physical curvature has been added, there is also NO guarantee that ANY grid points will lie at the location of the slope discontinuity!

How best to resolve a slope discontinuity :



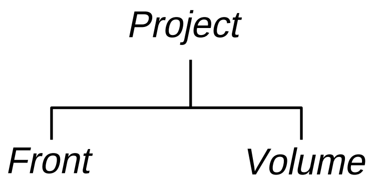
Since VGRID spline fits each curve (i.e., each edge of each side of each patch) individually, splitting the original curve into multiple curves (split at the location of the slope break) will guarantee that the spline will not "ring," and that VGRID will place a grid point at each end point.

Patch Shape is the Key.

The construction of "well−behaved" patches ALWAYS pays dividends later on in the grid generation process.

"Start−To−Finish" Outline
For Generating *Inviscid* Grids

Following the suggested directory structure :



1. Problem Set−Up **:**

➡ In the *"Front"* directory :

Using GridTool :

−Define patches
−Check for "Bad Patches" and account for anything that shows up
−Define background grid source (spacing) distribution
−Set "stretching" to zero (in the "Global" panel accessible from the "Background Grid" panel) *even if your eventual goal is to generate a mesh with stretching*
−Write GridTool restart file
−Write .d3m file (.mapbc file written automatically)

2. Initial* VGRID *run :
(to establish optimum patch orientation/side−ordering)

Using VGRID :

−Run VGRID interactively, specify project name as input
−Choose "Show when distorted triangles encountered" option from "Display surface triangulation on individual patches?" menu.
−Choose side−ordering that results in the best surface triangulation for each patch that VGRID flags as having "distorted" cells
−Choose "Write a new d3m file and stop" option (from the "Want to continue for volume grid generation?" menu) after having gone through all patches

(*NOTE :* this step only needs to be performed *once* for a given patch configuration. Subsequent modifications to the background grid source distribution *do not* require you to repeat this step, since the best patch orientation is purely a function of patch shape).

3. Ensuring compatibility between* VGRID .d3m *file and* GridTool *restart file :

Using GridTool :

−Read in GridTool restart file
−Highlight the VGRID .d3m file, and read it in as an "Update d3m" file
−Write out GridTool restart file

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

3. Ensuring compatibility between **VGRID .d3m** ***file and*** **GridTool** ***restart file*** , continued:

If you have sources with stretching :

Using GridTool :

−Set "stretching" to 1(in the "Global" panel accessible from the
"Background Grid" panel) to globally activate the stretching that you
have placed in the background grid sources (assuming that you wish to
implement stretching)
−Write out GridTool restart file
−Write out VGRID .d3m file

NOTE : If, in addition to reading in the .d3m file as an "update," any further modifications are performed (e.g., to patch shape or grid spacing controls), be sure to write a new GridTool restart file *and* a new VGRID .d3m file.

4. Generating the surface mesh :

Using VGRID :

−Run VGRID interactively, specify project name as input
−Choose the "No graphics" option from the "Display surface triangulation
on individual patches?" menu.
−Examine surface mesh using features in the "Plotting Options" menu once
the mesh on each patch is generated
−If surface mesh is adequate (cell face distribution and total mesh size), choose
the "Save the grid and stop" option from the "Want to continue for volume
grid generation?" menu. This will automatically write out all the surface mesh−
related files (the .bc, .front, and .cogsg files)
−if the surface mesh is inadequate, return to step 1 to refine either the patch
shape(s) or background grid source distribution (or both)

5. Generating the volume mesh :

→ Next, move to the *"Volume"* directory :

−copy (not move) the .d3m, .bc, .front, and .cogsg files over from the "*Front"*
directory (the GridTool restart file is not needed for this step) into the
"Volume" directory

The volume mesh can be generated interactively, *OR,* purely in batch mode, as described on the following page(s):

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

5. Generating the volume mesh, continued :

Interactive generation of the inviscid volume mesh using VGRID :

−Run VGRID, specify project name as input (specify "inviscid" for grid type)
−Hit "Esc" to bring up the display of the surface mesh
−Hit "Esc" once again to bring up the ’Plotting options" menu
−Choose the "Exit Plotting" option from the "Plotting options" menu
−From the "Want to continue for volume grid generation?" menu, choose the "Yes, and show the generation process on the fly" option, and select a rate at which VGRID will refresh the screen to display the newly−formed tetrahedra
−In order to reduce the total time spent generating the grid, it is recommended that you hit the "Ctrl" key to momentarily suspend the screen updates (this will turn the "fly" icon into a "running man" icon and disable the manipulation controls). Hitting the "Ctrl" key again will bring back the "fly" icon, enable the manipulation controls, and refresh the screen at the user−specified rate. Every time the "Ctrl" key is pressed, grid statistics (number of points, cells, etc.as of the time the key was pressed) are printed out in the window from which the VGRID session was launched.

NOTE : Any time that the "fly" icon is displayed, you can hit the "Esc" key and get the "Terminate/modify the generation process?" menu and choose to perform functions such as changing the screen refresh rate, changing the growth rate of the generated tetrahedral cells, or writing the entire mesh out (surface and volume, to the extent that it has been completed) for restarting (either interactively or in batch).

−Continue running until you get one of the following three messages (printed out in the window from which the VGRID session was launched):

a. *"Grid is complete. Please process the grid with POSTGRID"*
This indicates that the advancing−front algorithm has managed to fill the entire volume with tetrahedra, and that there are no points remaining on the "current front." Proceed to step 6 to attempt to improve the quality of the newly−completed mesh.

b. *"No more cells can be formed. Grid may now be completed by post−processing."*
This indicates that the advancing−front algorithm has filled the entire volume with as many tetrahedra as possible but that there remain areas ("pockets") where the algorithm was unable to insert nodes (and create cells) consistent with the user−specified spacing. Proceed to step 6 to complete the mesh.

c. *"Number of points generated exceeds maximum. Please increase mpoin."*
This indicates that the advancing−front algorithm has generated more points than is specified as the upper bound in the .d3m file (the "mpoin" value).
In this case, VGRID automatically writes all the grid−related files, thereby allowing you to restart the volume mesh generation (interactively or in batch) and generate another *mpoin* points. If you choose to restart, simply invoke VGRID and continue (VGRID will start from the place it stopped last, as indicated by the contents of the .front file).

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

5. Generating the volume mesh, continued :

NOTE : Regardless of which outcome you encounter (a, b, or c on the previous page), VGRID will automatically write the grid−related files out in the directory the session was launched from (the *"Volume"* directory in this case).

Batch generation of the inviscid volume mesh using VGRID :

−Run VGRID with the project name specified as a command line option. That is, type in

```
VGRID project_name
```

−The session will begin without any graphics or user input. VGRID will run until it encounters one of the three scenarios described earlier (see the messages a, b, and c listed under the interactive generation section). VGRID will then output one of the three messages listed earlier into a file which it will create and name *"project_name.info."* Again, VGRID will automatically write out the grid−related files in the directory from which the session was launched.

−Respond as appropriate : either proceed to step 6 (if you encountered messages "a" or "b"), or submit another VGRID run (interactive or batch) to complete the mesh (if you encountered message "c").

***6. Completing and improving the quality of the volume mesh using* POSTGRID** :

NOTE : This step *must* be performed even though VGRID may have completed the mesh (i.e., even though you encountered message "a" described section 5).

Using POSTGRID :

−Run POSTGRID interactively, specify project name as input
−Choose the proper mesh type ("inviscid")
−Choose "Unformatted (.cogsg) file" as the input grid file format

If the grid was complete prior to beginning the POSTGRID session :
(i.e., if you encountered message "a" described in section 5)

−POSTGRID will display the wireframe depicting the patch boundaries
−Choose from among the options listed in the "Options" menu that appears when you press "Esc." You can display the surface mesh (the "Surface mesh plot" option), display all the tetrahedra of the volume mesh that pass through user−specified constant−coordinate planes (the "Display of volume grid cut by a plane" option), or display individual (user−specified) tetrahedra in the mesh (the "Display of selected cells" option).

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
continued

6. Completing and improving the quality of the volume mesh using **POSTGRID** , continued :

If the grid was complete prior to beginning the POSTGRID session , continued:

−Choose one of the options available under the "Grid quality" menu item to perform a quality assessment and (if necessary) a "quality upgrade" pass.
−If POSTGRID determines that the mesh contains no distorted cells (on the basis of cell volumes, and, if the mesh has no stretching, on the basis of face angles), POSTGRID will respond with "no distorted cells found." In this case, POSTGRID will bring up the "Options" menu again, at which point you should write the files out by choosing the "Write the new data" menu item. You will most likely choose the "Unformatted (.cogsg) file" option (especially if you are going to be using USM3D as the flow solver). Should you need to transfer between formats (i.e., convert the .cogsg file to the .grd and .int, or vice−versa), utilities exist (POSTGRID itself is one) to perform the conversion. Once the grid is written out, you can exit POSTGRID. ***Proceed to step 7 (Projecting the surface triangulation to the surface definition).***
−If POSTGRID determines that the mesh contains distorted cells, it will list the cells and bring up the "Local Remeshing?" menu, from which you can register your choice as to whether or not to proceed with the remeshing of the grid in an attempt to reduce the number of distorted cells.
−If you chose to continue with the local remeshing, POSTGRID brings up the "Display of pockets (holes) desired?" menu. It is a good idea display these pockets, as they tend to give an indication of the areas where the grid contains distortion, and, potentially, where the grid generator had difficulty completing the mesh.
−After choosing "Yes" to display the pockets, POSTGRID updates the display to reflect the pockets (displayed in red), and allows you to manipulate the display to get a good idea as to their location in the domain. Once you are done inspecting the pockets, hit "ESC" to proceed.
−POSTGRID will then bring up the "Generation of volume grid on the fly?" menu, and allow you to specify a screen−refresh rate (under the "Yes" option), or to complete the mesh without any screen updates, or indeed quit the entire process if you see fit (under the "No" option).
−Assuming you chose to continue completing the mesh, POSTGRID will then attempt to complete the mesh by local remeshing. If you chose the "on the fly" option, you will actually see the pockets change as VGRID attempts to remesh (i.e., "fill") them.
−If POSTGRID is unable to complete the grid after one local remeshing pass, it will bring up another menu ("Removing layers of cells"), this time allowing you to choose the type and number of grid layers that you will permit POSTGRID to remove in it’s effort to form more "regularly−shaped" pockets and complete the mesh. Once you register your choice, POSTGRID will once again ask if you wish to view the pockets. After viewing them, hitting "Esc" will once again bring up the "Generation of volume grid on the fly?" menu. Proceed as before, letting POSTGRID attempt to complete the mesh. Every time POSTGRID is unable to complete the mesh, it will bring up the "Removing layers of cells" menu, allowing you to try another type of layer removal (or try the same type another time), and try again to fill in the pockets by locally remeshing. Hopefully, in the course of a few attempts, the grid will be completed, at which point POSTGRID will print out the "GRID IS COMPLETE" message and bring up the "Options" menu.

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

6. Completing and improving the quality of the volume mesh using **POSTGRID** , continued :

If the grid was complete prior to beginning the POSTGRID session , continued:

−The very first thing you should do when you encounter the "GRID IS COMPLETE" message is to write out the grid by choosing the "Write new data" menu item from the "Options" panel. You will most likely choose the "Unformatted (.cogsg) file" option (especially if you are going to be using USM3D as the flow solver). Should you need to transfer between formats (i.e., convert the .cogsg file to the .grd and .int, or vice−versa), utilities exist (POSTGRID itself is one) to perform the conversion.
−Once the grid is complete, you can once again choose from among the options listed in the "Options" menu. You can display the surface mesh (choose the "Surface mesh plot" option), display all the tetrahedra of the volume mesh that pass through user−specified constant−coordinate planes (the "Display of volume grid cut by a plane" option), or display individual (user−specified) tetrahedra in the mesh (the "Display of selected cells" option).
−Once the grid is written out, you can exit POSTGRID.
Proceed to step 7 (Projecting the surface triangulation to the surface definition).

If the grid was *not* complete prior to beginning the POSTGRID session :
(i.e., if you encountered message "b" described in section 5)

−POSTGRID will display the wireframe depicting the patch boundaries
−Upon hitting "Esc," POSTGRID will automatically proceed to attempt to complete the grid by local remeshing. It will open "pockets" around the areas where points remain on the front, and a menu will appear asking you if you want a display of the pockets that it has opened up. It is a good idea to display these pockets, as they tend to give an indication of the areas where the grid generator had difficulty completing the mesh, for whatever reason.
−After choosing "Yes" to display the pockets, POSTGRID updates the display to reflect the pockets (displayed in red), and allows you to manipulate the display to get a good idea as to their location in the domain. Once you are done inspecting the pockets, hit "ESC" to proceed.
−POSTGRID will then bring up the "Generation of volume grid on the fly?" menu, and allow you to specify a screen−refresh rate (under the "Yes" option), or to complete the mesh without any screen updates, or indeed quit the entire process if you see fit (under the "No" option).
−Assuming you chose to continue completing the mesh, POSTGRID will then attempt to complete the mesh by local remeshing. If you chose the "on the fly" option, you will actually see the pockets change as VGRID attempts to remesh (i.e., "fill") them.
−If POSTGRID is unable to complete the grid after one local remeshing pass, it will bring up another menu ("Removing layers of cells"), this time allowing you to choose the type and number of grid layers that you will permit POSTGRID to remove in it’s effort to form more "regularly−shaped" pockets and complete the mesh. Once you register your choice, POSTGRID will once again ask if you wish to view the pockets. After viewing them, hitting "Esc" will once again bring up the "Generation of volume grid on the fly?" menu. Proceed as before, letting POSTGRID attempt to complete the mesh.

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

6. Completing and improving the quality of the volume mesh using **POSTGRID** , continued :

If the grid was *not* complete prior to beginning the POSTGRID session , continued :

−Every time POSTGRID is unable to complete the mesh, it will bring up the "Removing layers of cells" menu, allowing you to try another type of layer removal (or try the same type another time), and try again to fill in the pockets by locally remeshing. Hopefully, in the course of a few attempts, the grid will be completed, at which point POSTGRID will print out the "GRID IS COMPLETE" message and bring up the "Options" menu. The very first thing you should do at that point is write out the grid by choosing the "Write the new data" menu item. You will most likely choose the "Unformatted (.cogsg) file" option (especially if you are going to be using USM3D as the flow solver). Should you need to transfer between formats (i.e., convert the .cogsg file to the .grd and .int, or vice−versa), utilities exist (POSTGRID itself is one) to perform the conversion.
−Once the grid is complete, you can once again choose from among the options listed in the "Options" menu. You can display the surface mesh (choose the "Surface mesh plot" option), display all the tetrahedra of the volume mesh that pass through user−specified constant−coordinate planes (the "Display of volume grid cut by a plane" option), or display individual (user−specified) tetrahedra in the mesh (the "Display of selected cells" option).
−Once the grid is written out, you could exit POSTGRID, since you have a completed mesh. It is recommended, however, that you attempt to improve the grid quality by choosing one of the options available under the "Grid quality" menu item to perform a quality assessment and (if necessary) a "quality upgrade" pass.
−If POSTGRID determines that the mesh contains no distorted cells (on the basis of cell volumes, and, if the mesh has no stretching, on the basis of face angles), POSTGRID will respond with "no distorted cells found." In this case, POSTGRID will bring up the "Options" menu again, at which point you should write the files out by choosing the "Write the new data" menu item. You will most likely choose the "Unformatted (.cogsg) file" option (especially if you are going to be using USM3D as the flow solver). Should you need to transfer between formats (i.e., convert the .cogsg file to the .grd and .int, or vice−versa), utilities exist (POSTGRID itself is one) to perform the conversion. Once the grid is written out, you can exit POSTGRID. ***Proceed to step 7 (Projecting the surface triangulation to the surface definition).***
−If POSTGRID determines that the mesh contains distorted cells, it will list the cells and bring up the "Local Remeshing?" menu, from which you can register your choice as to whether or not to proceed with the remeshing of the grid in an attempt to reduce the number of distorted cells.
−If you chose to continue with the local remeshing, POSTGRID brings up the "Display of pockets (holes) desired?" menu. It is a good idea to idea to display these pockets, as they tend to give an indication of the areas where the grid contains distortion, and, potentially, where the grid generator had difficulty completing the mesh.
−After choosing "Yes" to display the pockets, POSTGRID updates the display to reflect the pockets (displayed in red), and allows you to manipulate the display to get a good idea as to their location in the domain. Once you are done inspecting the pockets, hit "ESC" to proceed.

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

6. Completing and improving the quality of the volume mesh using **POSTGRID** , continued :

If the grid was *not* complete prior to beginning the POSTGRID session , continued :

−POSTGRID will then bring up the "Generation of volume grid on the fly?" menu, and allow you to specify a screen−refresh rate (under the "Yes" option), or to complete the mesh without any screen updates, or indeed quit the entire process if you see fit (under the "No" option).
−Assuming you chose to continue completing the mesh, POSTGRID will then attempt to complete the mesh by local remeshing. If you chose the "on the fly" option, you will actually see the pockets change as VGRID attempts to remesh (i.e., "fill") them.
−If POSTGRID is unable to complete the grid after one local remeshing pass, it will bring up another menu ("Removing layers of cells"), this time allowing you to choose the type and number of grid layers that you will permit POSTGRID to remove in it's effort to form more "regularly−shaped" pockets and complete the mesh. Once you register your choice, POSTGRID will once again ask if you wish to view the pockets. After viewing them, hitting "Esc" will once again bring up the "Generation of volume grid on the fly?" menu. Proceed as before, letting POSTGRID attempt to complete the mesh. Every time POSTGRID is unable to complete the mesh, it will bring up the "Removing layers of cells" menu, allowing you to try another type of layer removal (or try the same type another time), and try again to fill in the pockets by locally remeshing. Hopefully, in the course of a few attempts, the grid will be completed, at which point POSTGRID will print out the "GRID IS COMPLETE" message and bring up the "Options" menu.
−The very first thing you should do when you encounter the "GRID IS COMPLETE" message is to write out the grid by choosing the "Write new data" menu item from the "Options" panel. You will most likely choose the "Unformatted (.cogsg) file" option (especially if you are going to be using USM3D as the flow solver). Should you need to transfer between formats (i.e., convert the .cogsg file to the .grd and .int, or vice−versa), utilities exist (POSTGRID itself is one) to perform the conversion.
−Once the grid is complete, you can once again choose from among the options listed in the "Options" menu. You can display the surface mesh (choose the "Surface mesh plot" option), display all the tetrahedra of the volume mesh that pass through user−specified constant−coordinate planes (the "Display of volume grid cut by a plane" option), or display individual (user−specified) tetrahedra in the mesh (the "Display of selected cells" option).
−Once the grid is written out, you can exit POSTGRID.
Proceed to step 7 (Projecting the surface triangulation to the surface definition).

What should you do if you cannot complete the grid during the POSTGRID session? :

−There is a hard−coded limit of 10 attempts (to complete the mesh) per session within POSTGRID . Should you encounter the limit without a successful outcome (i.e., without POSTGRID having succeeded in completing the mesh), you should write out the grid (using the "Write the new data" menu item) and start a new POSTGRID session to try another series of attempts to complete the mesh.

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

6. Completing and improving the quality of the volume mesh using **POSTGRID** ,
continued :

If the grid was *not* complete prior to beginning the POSTGRID session , continued :

−If you observe that several POSTGRID sessions do not result in a completion of the mesh, and that the pockets that POSTGRID opens seem to be in the same location in the domain, chances are that there is something flawed in your surface mesh. In this situation, it is best to return to the *"Front"* directory and use either VGRID or GridTool to examine the surface mesh in detail (in particular, in the area where POSTGRID had difficulty completing the mesh). Once you determine a cause for the problem (which usually involves re−patching or making modifications to the background grid source distribution), start over from step 1.

7. Projecting the surface triangulation onto the surface definition :
(To obtain the projected surface mesh)

Next, move to the *"Front"* directory :

Before invoking GridTool, make a *copy* of the ".cogsg" file (which, in this directory, contains the coordinates and connectivity information for all the *surface* triangles). That is, create a copy that will retain the *un*projected surface mesh information, so that should you need to "undo" any projection−related errors later on in the process, you will have the unprojected data to "revert" back to. While you can name the copy anything, it is recommended that you name the copy something descriptive of it’s contents, such as "*project_name.cogsg_unp*."

Using GridTool :

−Read in GridTool restart file
−Highlight the .front file, and read it in as a "Front (VGRID)" format file.
−Proceed with projecting the surface triangulations on each surface patch onto the appropriate surface definition(s), as described in the GridTool User’s Manual.
−Write out the .front file (as a "Front (VGRID)" format file again) once you are done with the projections. This will overwrite the .front file to reflect the *projected* surface mesh coordinates. Even though it is a .front file you are writing over, GridTool will also write a new .cogsg file to reflect the projected surface mesh coordinates as well (hence the need to save a copy of the *un*projected .cogsg file for use if anything went wrong during the projection process).

8. Using **POSTGRID** ***to update the volume mesh to accommodate the projected surface mesh :***

NOTE : Obviously, by this stage in your mesh generation, you have made the judgement that the volume grid you have just completed is adequate (in terms of total number and distribution of point/cells). If for whatever reason you feel that the mesh is less than adequate for your particular problem, implementing changes to make the mesh better necessitates a return to Step 1, the "Problem Set−Up."

"Start−To−Finish" Outline
For Generating *Inviscid* Grids
(continued)

8. Using **POSTGRID** ***to update the volume mesh to accommodate the projected surface mesh*** , continued :

→ Next, move to the *"Volume"* directory :

Using POSTGRID :

−Run POSTGRID interactively, specify project name as input

−Choose the proper mesh type ("inviscid")

−Choose "Unformatted (.cogsg) file" as the input grid file format

−As during your last POSTGRID session, the code will begin by displaying the wireframe of the patch boundaries

−You can once again choose from among the options listed in the "Options" menu that appears when you press "Esc." You can display the surface mesh (the "Surface mesh plot" option), display all the tetrahedra of the volume mesh that pass through user−specified constant−coordinate planes (the "Display of volume grid cut by a plane" option), or display individual (user−specified) tetrahedra in the mesh (the "Display of selected cells" option. It is useful to choose the "Surface mesh plot" option so that you will be able to see the surface grid before and after projection (the projected and unprojected surface grids may not−−and really *should not*−−differ much, and as such, the changes introduced by "swapping" the unprojected surface with the projected one may not be visible, even if you could find a viewing orientation that would allow you to see all the patches)

−After displaying the surface mesh, hitting "Esc" brings up the "Options" menu again, from which you should now choose the "Moving Volume Grid" option. Upon choosing it (with a right−mouse click), the window from which you launched the session will appear, with the following instruction :

```
Please enter the file name for the moved surface ==> .front file
enter 0 to skip this option
```

−Specify the .front file that you created (using GridTool) in Step 7 (at the completion of the projection of the surface triangulations onto the surface definition). Note that you can supply the path to the .front file as well, so that you could type in

```
../Front/project_name.front
```

to tell POSTGRID where the file is located.

−POSTGRID will respond with

```
"constructing data structure..."
```

and then

```
"Moving the volume grid..."
```

"Start−To−Finish" Outline For Generating *Inviscid* Grids (continued)

8. Using **POSTGRID** ***to update the volume mesh to accommodate the projected surface mesh*** , continued :

−POSTGRID will then update the display to reflect the projected surface mesh (if you had the unprojected mesh displayed), and proceed to calculate the volume of each cell to ensure that no negative (crossed) cells have been created in the course of moving the grid to accommodate the projected surface mesh. If no such negative−volume cells were found, POSTGRID responds with

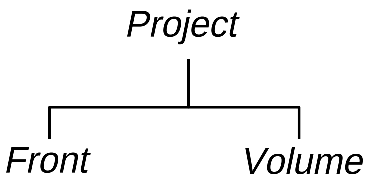
"No negative cells in inviscid portion"

You should now write out the grid (as before), or, if you are interested in a re−assessment of the grid quality, ask POSTGRID to perform another quality−assessment pass, as described earlier. Once you have written out the grid, you can exit POSTGRID. ***Your gridding is complete.***

−If, however, negative volumes have resulted from the grid movement, POSTGRID will automatically remove the problem cells and attempt to complete the mesh just as it does when you first invoked POSTGRID (see Step 6). In such a situation, POSTGRID cycles through as many local remeshing passes as needed (and as directed by user input, as before) to complete the mesh. Once the mesh is completed (signaled once again by the "GRID IS COMPLETE" message and the reappearance of the "Options" menu, you should write out the grid files. ***Your gridding is complete.***

"Start−To−Finish" Outline For Generating *Viscous* Grids

Following the suggested directory structure :



NOTE: Steps 1 − 4 for generating viscous meshes are identical to those for generating inviscid meshes, except for the following:

a. The "Viscous" parameter in the "Global" menu (accessible from the "Background Grids" panel) of GridTool should be set to 1.

b. The first layer spacing value ("Delta1," also accessible from the "Global" menu in the "Background Grids" panel of GridTool) should be set to the appropriate value (in dimensions consistent with the units of the grid).

c. The rates of grid layer growth "Rate1" and "Rate2" should be set to the appropriate values (again, these values are accessible from the "Global" menu in the "Background Grids" panel of GridTool).

d. All patches on which viscous layers are desired should have a viscous boundary condition associated with them (as set in GridTool) .

e. When generating the surface mesh (step 4), choose "viscous" as the grid type.

There is a utility code ("spc.f") that is useful in determining the Delta1 and Rate1 values, given the Reynolds number, the number of points desired in the boundary−layer, the y−plus value of the first point off the surface, and an estimate of the Rate2 value. This utility can be used for determining the Delta1 and Rate1 values whether or not you intend to use wall functions (i.e., whether or not the mesh you intend to generate will be designed to grid−resolve the boundary−layer).

The viscous grid generation sequence begins to differ from it's inviscid counterpart beginning with the generation of the volume mesh :

5. Generating the volume mesh :

➡ After generating the surface mesh, move to the *"Volume"* directory :

−copy (not move) the .d3m, .bc, .front, and .cogsg files over from the "*Front"* directory (the GridTool restart file is not needed for this step) into the *"Volume"* directory

The viscous volume mesh is generated in two parts : the advancing−layers portion, and the advancing−front portion. The advancing−layers portion *MUST* be generated interactively.

"Start−To−Finish" Outline For Generating *Viscous* Grids (continued)

5. Generating the volume mesh , continued**:**

Interactive generation of the advancing−layers portion of the volume mesh :

−Run VGRID, specify project name as input
−Specify "viscous" as the grid type
−Hit "Esc" to bring up the display of the surface mesh
−Hit "Esc" once again to bring up the ’Plotting options" menu
−Choose the "Exit Plotting" option from the "Plotting options" menu
−From the "Want to continue for volume grid generation?" menu, choose the "Yes, and show the generation process on the fly" option, and select a rate at which VGRID will refresh the screen to display the newly−formed tetrahedra
−VGRID will bring up the window from which the session was launched, and print out the message

```
Preparing data structure and surface vectors...
```

indicating that VGRID has begun setting up the information needed to generate the advancing−layers. This "setting up" is comprised of two parts :

a. pre−determining all the proper tetrahedron "types" to ensure that adjacent groupings of cells within a layer are compatible (refer to AIAA Journal article "Three−Dimensional Unstructured Viscous Grids by the Advancing−Layers Method" for details). Once VGRID completes the determination of the connectivities/data structure, it prints out the message

```
cell connectivities OK
```

and proceeds to the second step of the "setting up," which is...

b. computing the vectors along which the grid points (that form the viscous grid layers) are going to be generated. These vectors emanate from each node on the surface mesh that lies on a patch marked as having a viscous boundary condition (as set in GridTool), and must satisfy the so−called "visibility" criterion (as described in AIAA Journal article "Three−Dimensional Unstructured Viscous Grids by the Advancing−Layers Method"). VGRID then prints out the message

```
computing surface vectors. Please wait...
```

to indicate that it is calculating the vectors. If all the vectors satisfy the "visibility" criterion, VGRID returns the message

```
validity of surface vectors confirmed.
```

and brings up the menu asking if you would like to see a plot of the surface vectors. In general, if VGRID has confirmed the validity of the vectors, there is no real reason to view them (other than curiosity).

"Start−To−Finish" Outline For Generating *Viscous* Grids (continued)

5. Generating the volume mesh, continued :

Interactive generation of the advancing−layers portion of the volume mesh :

Should you choose to view them, simply hit "Esc" after examining them to signal VGRID to proceed with the generation of the advancing−layers. If you choose not to view the vectors, registering a "No" will automatically start the generation of the advancing−layers. In either case, the message window will disappear, and the screen will change color, accompanied by the appearance of the "fly" icon, as during the inviscid mesh generation.

What will happen if VGRID finds "invisible" vectors? :

−If, during the determination of the surface vectors, VGRID prints out the message

error! invisible vector detected

this indicates that there is at least one surface vector that cannot satisfy the "visibility" criterion. VGRID will then allow you to view the surface vectors, (by clicking "yes" on the "Plot of surface vectors desired?" menu). Hitting the "t" key will toggle between several renderings of the surface mesh and the vectors, allowing you to examine the location of the "invisible" vectors (which will be rendered in red). Once you are done examining them, hitting "Esc" will terminate the VGRID session (since the advancing−layers algorithm cannot proceed without a complete and valid set of surface vectors). Upon terminating the session, VGRID will also list a few of the probable causes for the presence of the invalid vectors. The most likely causes have their roots in a poor surface mesh (as with just about any problem that results in VGRID failing).

There are cases, however, in which certain features of the geometry are such that no valid vector(s) can be found emanating from the nodes of the surface triangles in the area in and/or around the feature. In these cases, the user must introduce slight modifications to the geometry in order to allow VGRID to determine valid vector(s). In such cases, the user must return to step 1, and perform the modification to the patching.

Assuming that all the surface vectors were valid, you proceed as follows :

−The "fly" icon appears to indicate that VGRID is generating tetrahedra using the advancing−layers method (i.e., distributing points along the surface vectors according to the user−specified Delta1, Rate1 and Rate2, and stacking groupings of cells one layer at a time).

"Start−To−Finish" Outline
For Generating *Viscous* Grids
(continued)

5. Generating the volume mesh, continued :

Interactive generation of the advancing−layers portion of the volume mesh :

−In order to reduce the total time spent generating the grid, it is recommended that you hit the "Ctrl" key to momentarily suspend the screen updates (this will turn the "fly" icon into a "running man" icon and disable the manipulation controls). Hitting the "Ctrl" key again will bring back the "fly" icon, enable the manipulation controls, and refresh the screen at the user−specified rate. Every time the "Ctrl" key is pressed, grid statistics (number of points, cells, etc.as of the time the key was pressed) are printed out in the window from which the VGRID session was launched.

NOTE : Any time that the "fly" icon is displayed, you can hit the "Esc" key and get the "Terminate/modify the generation process?" menu and choose to perform functions such as changing the screen refresh rate, etc. Currently, the generation of the advancing−layers *MUST* be performed in a *single, interactive* session of VGRID. Hence, you *cannot* generate the advancing−layers in several restarts (as you can with grids generated using the advancing−front algorithm alone).

−Upon completion of the advancing−layers, VGRID will calculate the volumes of each cell generated thus far, and locate any that are negative (i.e., that have "crossed" with adjacent cells). If any negative volume cells are found, a menu comes up allowing you to view a display of the cells (register your choice with a right−mouse click, as always). It is a good idea to view the location of the negative cells, as it can be an indication of the areas where VGRID might have difficulty completing the mesh.

−After examining the negative cells, hitting "Esc" will prompt VGRID to ask if you would like to save the viscous grid and stop. At the current stage of the code’s development, it is recommended that you choose "yes." VGRID then writes all the appropriate grid files and terminates execution. The window from which you launched the session will now show a summary of the grid statistics for the (advancing−layers) portion of the grid generated thus far.

Generation of the advancing−front portion of the (viscous) volume mesh :

−The remainder of the volume can now be gridded using the pure advancing−front algorithm, such as was used to generate the entire inviscid mesh. As such, you are free to run VGRID in either interactive or batch modes, just as described in step 5 of the inviscid mesh generation procedure. If you choose to run VGRID interactively, however, be sure to specify "viscous" as the grid type, even though you are generating the advancing−front (i.e., the "inviscid") portion of the mesh.

"Start−To−Finish" Outline
For Generating *Viscous* Grids
(continued)

5. Generating the volume mesh, continued :

Interactive generation of the advancing−layers portion of the volume mesh :

−As with the generation of an inviscid mesh, VGRID will run until it encounters one of the scenarios described in step 5 of the inviscid volume mesh generation procedure.
−Respond as appropriate (as described in step 6 of the inviscid volume mesh generation procedure), with the eventual goal of proceeding to POSTGRID.

6. Completing and improving the quality of the volume mesh using **POSTGRID** :

NOTE : This step *must* be performed even though VGRID may have completed the mesh.

Using POSTGRID :

−Run POSTGRID interactively, specify project name as input
−Choose the proper mesh type ("viscous")
−Choose "Unformatted (.cogsg) file" as the input grid file format

Again, the sequence for completing and improving the quality of the viscous (i.e., advancing−layers+advancing−front) grid is identical to that for generating purely inviscid (advancing−front) grids, except that now, you may avail yourself of the option to remove 1 layer of viscous (advancing−layer) cells in addition to inviscid (advancing−front) cells in areas where the mesh is trying to locally remesh to complete the volume.

NOTE : Once the volume mesh is completed, the remaining steps in the sequence (involving projecting the surface mesh onto the original surface definition using GridTool, and then "moving" the newly−generated volume mesh to accommodate the projected surface mesh) are identical to those in the inviscid mesh sequence. Please refer back to steps 7 and 8 of the inviscid mesh sequence for the details.

The VGRID 3.2 Sequence in More Detail:

I. Generating a surface mesh (the "initial front") interactively : (Step II in the "Event Sequence")

File you need : 1. the .d3m file

Sequence of events : 1. Invoke VGRID

2. Specify the "project name" when VGRID asks "Enter project name for an interactive run:"

3. Specify (via a right−mouse click) whether the grid to be generated is inviscid (i.e., pure advancing front) or viscous (i.e., advancing layers + advancing front). NOTE : the choice made at this stage will override the settings specified in the .d3m file.

4. Once the wireframe of the configuration (as delineated by the patch boundaries in the .d3m file) appears, hit "ESC" to bring up the next menu, which will ask if you wish to see the placement of the background grid sources. Note that only the source locations and numbers (and NOT the spacings or intensity factors, etc) will be displayed. Register your choice with a right−mouse click.

5. Hit the "ESC" key to proceed to the next menu, which will ask if you wish to see a display of the points on the patch boundaries. Note that the points that VGRID will display at this stage correspond to points that will actually form the nodes of the surface triangles (on the boundaries of the user−defined surface patches). As such, it is good practice to examine the boundary grid point distribution to get a general idea of whether the grid points are distributed in the manner expected (in the course of setting up the background grid sources in GridTool). If, for instance, it is observed that the grid is too coarse or fine in certain areas of interest, or the nodes do not reflect the desired/expected stretching, the user can decide to quit the session and return to GridTool to redo the background grid source distribution. Hitting "ESC" will bring up the menu that will prompt you for your decision (to proceed with the surface mesh generation, or quit, as indicated by the "Display surface triangulation on individual patches?" menu). Again, a right−mouse click registers your choice. Details pertaining to the different choices are discussed in the next step of the sequence.

6. If you decide to continue to the surface generation stage, you have the option of having VGRID generate the surface triangles on each surface patch without updating the screen (the "No Graphics" option). However, during your very first interactive run (i.e., after having completed the patch construction and source placement in GridTool), it is highly recommended that you allow VGRID to detect and flag

I. Generating a surface mesh (the "initial front") interactively, CONTINUED :

any surface triangles (on 4−sided patches) that are highly skewed (i.e., beyond a certain percentage of being equilateral). This option is selected by choosing (via a right−mouse click) the "Show when distorted triangles encountered" menu option. VGRID will then generate the surface triangulation on each patch, and stop to display a particular 4−sided patch only when distorted triangles are encountered. VGRID then allows the user to re−order ("rotate") the 4−sided patch on which skewed triangles are detected in order to determine which orientation yields the smallest number of skewed cells. More detail on the "patch rotation" process can be found in the next steps of the sequence.

NOTE: The option of allowing VGRID to detect skewed cells is only available if the global stretching switch in the .d3m file (the "strch: 0 off; 1 on" line in the "Parameters" section of the file) is set to 0 (i.e., stretching is turned off). Even if your eventual goal is to generate a (viscous or inviscid) mesh incorporating some stretching, it is recommended that (at least for your very first run through VGRID), you temporarily suppress the implementation of the stretching parameters specified in the .d3m file to allow VGRID to detect skewed triangles. If stretching is not suppressed, choosing any option (either the "Show triangulation on a selected patch" or "Show when distorted triangles encountered" options) will result in VGRID generating the entire surface grid without checking for skewed triangles and updating the screen, as in the "No graphics" option. After you have gone through the "patch rotation" process once (and have saved the best orientations in the .d3m and restart files), you can perform all subsequent surface generations using the "No graphics" mode (which will, of course, be the default mode if the global stretching switch is set to 1 in the .d3m file).

7. Having chosen the "Show when distorted triangles encountered" option, VGRID will generate the surface mesh on each of the patches until a 4−sided patch containing at least one skewed cell is encountered. VGRID will then display the patch, highlighting the distorted cells in red, and allow you to examine (i.e., rotate, translate) it. Hitting the "ESC" key brings up the "Select a Patch" menu, from which you can find the patch number of the currently displayed patch by locating the first patch number that does not have an asterisk (*) next to it. A right−mouse click on that patch number will result in a display of the same patch, but with a side rotation having been performed. That is, the ordering of the sides has "shifted" so that the patch side that was side number 2 becomes side number 1, and so on (the side that was side number 4 becomes side number 1). The image on the screen depicts the patch shape in computational space (where all VGRID's surface

generation takes place). Hitting the "ESC" key brings up the "Want to triangulate this patch?" menu, from which you can choose to view the advancing−front algorithm work "on the fly" (not recommended unless you are debugging a problem with a particular patch), or to view the final result of the triangulation on that particular patch. Registering your choice with a right−mouse click, the final triangulation will then be displayed in red, and still in computational space. Hitting "ESC" will then display the same patch in physical space, allowing you to see if the side re−ordering resulted in fewer skewed cells. Hitting "ESC" again will bring up the "Select a patch" menu again, allowing you to re−do the existing patch (i.e., try another side re−ordering), or proceed with the surface generation until another patch with skewed cells is located (by choosing the "Exit plotting" option).

NOTE : If, at any time during the surface mesh generation, you decide (based on the grid quality and/or density of the surface mesh) that either the patching or the background grid source distribution is inadequate and must be redefined, you can choose the "Exit" option from the "Select a Patch" menu and exit VGRID with or without saving any patch rotations that have been performed onto the .d3m file. At the current stage of the code’s development, however, you cannot save a partially−generated surface mesh and re−start VGRID to complete it later (as you can during the volume mesh generation).

8. After having generated the surface mesh on all the surface patches, and having rotated the patches on which VGRID detected skewed surface triangles to obtain the patch orientation yielding the "best" surface triangulation, VGRID displays the entire surface mesh and allows you to examine (i.e., translate, rotate) it. Hitting "ESC" brings up the "Plotting Options" menu, from which you can choose several ways of displaying the surface mesh (or just the outline of the surface patches, or just the distribution of background grid sources, etc.). Given that you do not know ahead of time how many points VGRID will generate in the course of completing a volume mesh, it is a good idea to use the "Distribution of grid on a plane in the field" option (from the "Plotting options" menu) to allow VGRID to display a very accurate representation of the grid point distribution on a plane through the field. Keep in mind that while VGRID has not (at this stage) generated any volume elements, it has all the information it needs (i.e., the same background grid sources that were used to determine the size of the surface triangles) to do so. As such, VGRID allows you to view the results of generating a mesh on a planar cut through the volume using the same spacing information that it will actually use to generate the volume mesh. The result is a very accurate representation of the density and growth rate of the volume mesh that will result from the background grid source distribution. Examining it allows

you to make a judgement as to whether the resulting volume mesh will be too dense or too coarse. After choosing the constant−coordinate plane on which to examine the mesh (and registering your choice with a right−mouse click), VGRID computes the planar mesh, then displays it at a default location (the average of the min and max coordinate values in the chosen direction). You can move the location of the plane by pressing the right and left arrow keys. Once you move the plane, only the green−colored representation of it is shown. Once you place it at the location of your choice, hitting the "ESC" key tells VGRID to compute the planar grid at that location and display it. Note that the grid point distributions displayed using this option do not reflect the presence of any part of the grid that will be generated (later) using the "advancing layers" method. The planar grids do, however, however, reflect any stretching that you may have implemented in your background grid source distribution. Hitting "ESC" again brings up the menu from which you can choose to examine the grid distribution on other constant−coordinate planes, or return to the "Plotting options" menu.

9. Examining the surface mesh (along with the representations of what the resulting volume mesh will be like) leads you to a decision as to the adequacy of the mesh, and therefore, whether you will proceed with generating the volume mesh, or discard the surface mesh and return to GridTool to make improvements.

   Choosing the "Exit Plotting" option from the "Plotting options" menu brings up the next menu, which asks "Want to continue for volume grid generation?" While you can certainly proceed with the generation of the volume mesh (choosing either of the "Yes" options to generate the mesh with or without updating the screen graphics), it is recommended that you terminate the session and save the appropriate data. Use the following as a guide:

   a. If you were running VGRID for no other reason than to obtain the best orientation for each patch (by allowing VGRID to flag skewed surface triangles), choose the "Write a new .d3m file and stop" option. You will then read the "new" .d3m file (containing the chosen patch orientations as decided during the VGRID session) back into GridTool as an "update" .d3m file (see "Event Sequence," Part IV).

   b. If you were running VGRID to generate the surface mesh that you will actually use (i.e., to get the volume mesh), AND the surface mesh you just generated meets your requirements for grid density and quality, choose either the

      "Save the grid and stop " option IF no patch rotations were performed in the course of generating the surface mesh, OR

I. Generating a surface mesh (the "initial front") interactively, CONTINUED :

choose the "Save both the grid and a new .d3m file" option IF any patch rotations were performed in the course of generating the surface mesh.

c. If you are unhappy with the grid and decide that you need to return to GridTool to redefine patches or background grid sources, you can choose to discard the surface mesh by choosing the "Stop! Discard the new data" option.

Assuming that you didn't choose "c," you now have your surface mesh. If any patch rotations were performed in the course of generating the surface mesh, it is always good to invoke GridTool, read in your restart file, then your "new" .d3m file as an "update d3m" file. Writing the restart file back out will then guarantee that the patch orientations decided upon during the surface mesh generation stage are now reflected in the restart file. This is important, since any future modifications you might introduce to the patching or spacing distribution (which will of course necessitate the generation of a new surface mesh) will guarantee that the 4−sided patches are oriented in the best possible way. Remember that for a given patch arrangement, you need only go through the "patch re−orientation" process once. The best orientation is purely a function of patch shape. Of course, if you observe that none of the 4 possible patch orientations yields a smooth, well−behaved mesh, there is a good likelihood that your basic patch shape is suspect (and the patch probably falls into the "bad" or "very bad" categories as shown on the page entitled "Some examples of good and bad patches for VGRID").

One might want to go ahead and take the time to project all the surface triangles onto the appropriate underlying surfaces (using GridTool) at this point. If, however, you do not already have an estimate of the total number of points/cells that will result from the volume grid generation process at this point (say, from a previous volume generation of an earlier similar case), it is recommended that you proceed to generate the volume, and project only after the volume has been completed.

It is recommended that you copy all the surface−mesh related files (the .d3m, the .bc, the .front and .cogsg files) onto another directory (as outlined in the page entitled "Suggested directory structure for generating grids with VGRID") to prevent any errors (from any source) from affecting the surface mesh. Also, once a satisfactory volume has been generated, the user can return to the surface mesh directory, perform the appropriate projection of the surface mesh, and then use the projected mesh to "update" the volume mesh, using the "Moving Grid" option in POSTGRID (discussed later).

II. Generating a volume mesh interactively :

Files you need : 1. .d3m
2. .front
3. .cogsg
(NOTE : the .bc file is "frozen" after the surface mesh generation)

Sequence of events : 1. Invoke VGRID

2. Specify the "project name" when VGRID asks
"Enter project name for an interactive run:"

II. Generating a volume mesh interactively , CONTINUED :

NOTE : VGRID first checks for the existence of a .front file in the directory in which you are running. If it finds one (with the prefix you specified as the "Project name" in step 2), then it assumes that you already have a surface mesh, and wish to proceed to generate the volume mesh.

3. Specify (via a right−mouse click) whether the grid to be generated is inviscid (i.e., pure advancing front) or viscous (i.e., advancing layers + advancing front). NOTE : the choice made at this stage will override the settings specified in the .d3m file.

4. Once the wireframe of the configuration (as delineated by the patch boundaries in the .d3m file) appears, hitting "ESC" brings up a display of the surface mesh. Hitting "ESC" again brings up the "Plotting options" menu. This is the same menu that was available at the completion of the surface mesh generation stage (step 8 of part I), which allowed you to render the surface in different ways (surface patches only, shaded surface, etc.) as well as allowed you the option of letting VGRID display an accurate representation of the volume mesh on constant−coordinate planes in the field. Choosing the "Exit plotting" option (with a right−mouse click) brings up the "Want to continue for volume grid generation?" menu (as in step 9, part I). This time, however, you should choose one of the first two options (generating the grid with or without periodically updating the screen). You will most frequently choose the first option (the "Yes, and show the generation process on the fly" option), and specify (with a right−mouse click) the rate at which the screen will be updated to reflect the "growth" of the mesh as the advancing−layers or advancing−front adds points to the field to form tetrahedra. If you are generating a viscous mesh, note that you *must* choose this option. VGRID’s response to choosing the "Yes, and show the generation on the fly" option (and selecting a screen update frequency) will depend on whether you are generating a viscous or inviscid mesh:

   a. If you are generating an inviscid mesh, the screen will change color (to a light green, as opposed to the gray used for rendering the surface mesh), and the advancing−front algorithm will immediately begin generating field points (in accordance with the user−specified background grid spacing) and forming tetrahedral cells. VGRID will update the screen at the user−specified rate, and you will actually see the front advancing and slowly filling up the volume with cells. You will observe that VGRID begins creating cells in the areas of the mesh where the cell spacings are smallest. You will also notice the fly−shaped cursor on the screen, indicating that the mesh is being generated "on the fly".

II. Generating a volume mesh interactively , CONTINUED :

Hitting the "CTRL" key at any time temporarily suppresses the screen updates (i.e., just as though you had chosen the "no graphics" option earlier), and allows VGRID to generate the mesh more rapidly. It will also replace the "fly" cursor with that of a "running man," indicating that VGRID is indeed running in the background, despite the fact that it is not updating the screen to display the generated cells. Note that all object manipulation keys are also suspended when in this "running man" mode. Hitting "CTRL" again returns you to screen−update mode, and once again enables object manipulations (and brings back the "fly" cursor). Note that every time "CTRL" is hit, summary statistics of the evolving grid (in terms of number of points and tetrahedral cells, as well as current rate of grid growth as described in item ii below) are printed out in the window from which you launched the VGRID session.

At any time during the "on the fly" advancing−front volume mesh generation, hitting "ESC" will bring up the "Terminate/modify the generation process?" menu, from which you can choose to:

i. Change the screen−update rate (the "Change the rate of graphics update" option)

ii. Increase or decrease the rate of the advancing−front method (AFM) grid growth. Clicking (with the right−mouse) on either of these will result in VGRID generating cells that are 5 % "taller" (if you chose to increase the rate) or 5 % "shorter" (if you chose to decrease the rate) than the background grid dictates. Each time this option is chosen, the rate is increased or decreased by 5 %. There is, however, a hard−coded limit on the total amount by which the growth rate can be increased. The current rate of grid growth is printed out periodically in the window from which you launched the VGRID session. Keep in mind that changing the grid growth rates requires you to "babysit" the grid as it is being generated, since you must be sure to ramp the growth rate back to 1 as the volume mesh nears completion to avoid large disparities between the cells being generated and the surface triangles on the outer boundaries. As such, it is an option useful mainly when you are trying to reduce the total cell count (especially further out in the field away from the configuration surface).

iii. Terminate the process, either for purposes of restarting the volume mesh generation purely in batch (an option not available for the advancing−layers portion), or discarding the mesh entirely.

b. If you are generating a viscous mesh, the window from which you launched the VGRID session will appear, and the message

```
Preparing data structure and surface vectors...
```

will appear. At this point, VGRID is determining the way in which each cell that will be generated in the viscous layers must be arranged (within each grouping of three tetrahedra that comprises each "prism−like" element in each layer) so that the diagonals "match up" between adjacent cell groupings (refer to pages 44−45 of the AIAA Journal article entitled "Three−Dimensional Unstructured Viscous Grids by the Ad− Vancing Layers Method"). VGRID will use this arrangement of cells throughout the advancing−layers portion of the mesh generation. VGRID iterates until it finds a valid set of cell connectivities, which it indicates with the message

```
cell connectivities OK
```

in the message window. Next, VGRID begins the computation of the vectors along which it will distribute grid points in the advancing−layers portion of the mesh (again, refer to the AIAA Journal article mentioned above), as indicated by the message

```
computing surface vectors. Please wait...
```

If VGRID determines that all the vectors emanating from each node of each surface triangle on a patch having a viscous boundary condition has met the so−called "visibility condition" (as explained in the above−mentioned AIAA Journal article), then the message

```
validity of surface vectors confirmed.
```

appears in the message window. If one or more "invisible" vectors (i.e., vectors that have failed the "visibility condition") are detected, then VGRID prints out the message

```
error! Invisible vector detected
```

In either case (whether VGRID detected any "invisible" vectors or not), a menu now appears allowing you to view a plot of the surface vectors if you wish. Obviously, if there were any "invisible" vectors determined, you want to find out exactly where they were located. Hitting the "t" key will toggle between a display of the vectors on the wireframe of the surface mesh, a display of the vectors on a shaded surface, and a display of the vectors on the wireframe of the patch outline.

If any "invisible" vectors have been found, hitting "t" will display only those vectors, highlighted in red. Hitting "ESC" will then terminate the VGRID session, since the presence of the problem vectors will prevent a valid advancing−layers mesh from being generated. VGRID will also make a few suggestions as to what the possible causes for the problem might be. As with just about any problem that VGRID can encounter, the cause can almost always be traced to a poor surface mesh.

Assuming that all the surface vectors were deemed valid, hitting "ESC" then signals VGRID to begin generating the advancing−layers portion of the mesh. As with the advancing−front portion, the screen will be updated at the user−specified frequency, and the "fly" cursor will appear. Also, you can once again suspend the screen updates by hitting "CTRL", and resume them by hitting "CTRL" again. The number of cells generated per layer, along with a running count of the total number of cells in the mesh is printed out in the window from which you launched the session.

Upon completion of the advancing−layers, VGRID will calculate the volumes of each cell generated thus far, and locate any that are negative. If any cells with negative volume are found, a menu comes up allowing you to view a display of the cells. As always, register your choice with a right−mouse click.

VGRID will then ask if you would like to save the viscous (i.e., advancing−layers portion) of the mesh and stop. At the current state of the code’s development, it is recommended that you choose "yes." VGRID then writes the appropriate files, and terminates execution. The window from which you launched the session will now show a summary of the grid statistics for the portion of the grid generated thus far.

5. If you are generating an inviscid mesh, then VGRID will simply continue with the advancing−front until it either completes the volume, or cannot insert any more nodes consistent with the spacing dictated by the background grid. Either way, the next step is to complete and/or improve the mesh with the POSTGRID code.

   If you are generating a viscous mesh, the remainder (i.e., the advancing−front portion) of the mesh is generated just as though you were generating an inviscid mesh. You begin by invoking VGRID (in the same directory containing the files just generated by the advancing−layers part of the process). You then specify "viscous" (despite the fact that you are generating the "inviscid" portion of the mesh). VGRID will then display the wirferame of the patch

II. Generating a volume mesh interactively , CONTINUED :

boundaries. Hitting "ESC" will bring up a display of the mesh, complete with the layers generated by the advancing−layers method. This is now essentially the "initial front" from which the advancing−front method will begin to fill in the rest of the volume. Hitting "ESC" again will bring up the by now familiar "Plotting options" menu, from which you will most frequently select "Exit plotting" to prompt VGRID to proceed with the volume mesh generation.

At this stage, you are essentially back to step 4, and can, in fact, follow the steps given in part "a" (for generating an inviscid mesh). VGRID will then run until it either completes the mesh (the "Grid is complete" message) or cannot insert any more nodes consistent with the spacing dictated by the background grid (the "No more cells can be formed" message). VGRID will then terminate execution, and summarize the grid statistics in the display window.

6. The final step for either viscous or inviscid meshes is to run POSTGRID (in the same directory containing the volume mesh just generated by VGRID). Invoke POSTGRID, and specify the project name, as done previously with VGRID. Note that you MUST run POSTGRID, regardless of the fact that VGRID may have printed out the "Grid is complete" message at the termination of the advancing−front grid generation stage described in step 5 above.

7. Specify if the mesh is viscous or inviscid with a right−mouse click.

8. Specify the format of the grid you are reading in. Most frequently, you will be choosing the "Unformatted (.cogsg) file" option.

9. POSTGRID will read in the grid files, and determine if the grid requires completing (by local remeshing) or is in fact, complete, in which case it is ready to be written out and/or improved in quality.

   If the grid requires completing (the more likely scenario), a menu will appear allowing you to choose how you would like POSTGRID to remove layers of cells surrounding the "unclosed" portions of the grid. There is no established/preferred set of choices, but it is generally preferable to try and complete the mesh while minimizing the number of cells that must be removed from the viscous layers.

10. After registering your choice, POSTGRID will ask if you wish to display the "holes" that it is attempting to fill with tetrahedra. You will most frequently choose "yes," as this will allow you to locate the areas where the grid is having difficulty completing, which is always instructive to know.

II. Generating a volume mesh interactively , CONTINUED :

11. POSTGRID will then display the "holes" in red, allowing you to see where they are in the field. Hitting "ESC" again brings up the menu from which you can choose to view the mesh generation (within the "pockets") on the fly.

12. POSTGRID will then attempt to complete the mesh within the pockets, and continually bring up menus with more cell removal options if it continues to encounter difficulty. You then choose an option, and see if results in the "GRID IS COMPLETE" message. There is a fixed upper limit for the number of attempts one can make to complete the mesh. If this limit is reached, POSTGRID prints out the "SORRY, GRID CANNOT BE CLOSED" message. If this occurs, be aware that you can write out the grid "as is," and restart POSTGRID for another series of attempts.

13. If you have completed the mesh, it is always best to choose the "write the new data" option and thus save your newly−generated mesh. This guarantees that any errors/problems that might be introduced in the course of attempting to improve the mesh will not affect your completed mesh.

14. Once you have completed the mesh, you can view the volume by choosing the "Display of volume grid cut by a plane" option. This will show the actual cells (from both the advancing−layers and advancing−front portions of the mesh) cut by a plane whose position you specify (as with the plane described in step 8, part I).

15. You can attempt to improve the quality of the cells in the advancing−front portion of the mesh by choosing the "Grid Quality" option (or simply ask for a grid quality assessment). Choosing any of the improvement options will result in cell removal (in the areas around the "low quality" grid cells), and will therefore require that you complete the mesh again, as in steps 10 − 12.

16. Always save any completed mesh that results from a quality−improvement pass.

III. "Updating" the volume mesh to accommodate the projected surface mesh:

(i.e., using the "Moving Volume Grid" option in POSTGRID)

Assuming that you are satisfied with the quality of the volume mesh (i.e., the total number of cells, their distribution, and deviation from the user−specified cell volumes as summarized during a "Grid Quality Assessment" pass in POSTGRID) that you have just generated, you can now use the "Moving Grid" option in POSTGRID to "swap" the unprojected surface mesh that you used (from which the advancing layers/front advanced to fill the volume and form your volume mesh) with the projected surface mesh.

III. "Updating" the volume mesh to accommodate the projected surface mesh:

(i.e., using the "Moving Volume Grid" option in POSTGRID), continued.....

1. Begin by returning to the directory where all the surface−mesh−related files were generated (the "Front" directory if you are following the "suggested" directory structure outlined in this document), and invoke GridTool. Read in your GridTool restart file, and then read in the .front file as a "Front (VGRID)" format file (as described in step III of the "Event Sequence" and step III of the "Input/Output Files" section of this document).

2. Project the surface triangulations onto the appropriate underlying surface definitions as described in the GridTool manual.

3. Upon completion of the projection process for all the patches requiring projection, write out a new .front file, which will contain the projected (x,y,z) coordinates of all the nodes on the surface triangulation. Please note the suggestions listed in step III of the "Input/Output Files" section of this document pertaining to taking the prudent step of saving the "unprojected .cogsg" file prior to writing the projected front file.

4. Return to the directory containing the recently−completed volume mesh (the "Volume" directory if you are following the "suggested" directory structure outlined in this document) and invoke POSTGRID. Specify the project name, and the format of the grid files to be read in (usually the unformatted .cogsg file).

5. As with the last time you were in POSTGRID, the code will begin by displaying the wireframe of the patch boundaries. Hitting "ESC" will bring up the usual "Options" menu. At this point it is useful to choose the "Surface Mesh Plot" option so that you will be able to see the grid before and after the projected mesh has been read in and "swapped" with the unprojected mesh (the two may not−−and really *should* not−−differ much, and as such, the changes introduced by the "swap" may not be visible, even if you could find an orientation that would allow you to see all the patches).

6. After displaying the surface mesh, hitting "ESC" again brings up the "Options" menu again, from which you should now choose the "Moving Volume Grid" option. Upon choosing it (with a right−mouse click), the window from which you launched the session will appear, with the following line :

   Please enter the file name for the moved surface mesh ==> .front file
   enter 0 to skip this option.

   At this point, you should specify the .front file that GridTool just wrote in step 3 above . Note that you can supply the path to the .front file as well, so that if you are following the "suggested" directory structure outlined in this document , you might type in

   ../Front/project.front

7. POSTGRID will respond with

   "constructing data structure..."

   and then

   "Moving the volume grid..."

III. "Updating" the volume mesh to accommodate the projected surface mesh:

(i.e., using the "Moving Volume Grid" option in POSTGRID), continued.....

POSTGRID will then calculate the volume of each cell (in both the advancing−layers and advancing−front portions of the mesh) to ensure that no negative (crossed) cells have resulted from the grid movement introduced by the projected surface mesh. If no such negative cells were found, POSTGRID responds with

"No negative volumes in inviscid portion"

and

"No negative cells in viscous portion" (if your mesh had any advancing−layers)

You should now write the grid files out (as before), and, if you are interested in the grid quality, ask POSTGRID to perform a quality assessment pass.

If, however, negative volumes resulted from the grid movement, POSTGRID will automatically remove the problem cells and attempt to complete the mesh just as it did when you first invoked POSTGRID (after VGRID indicated that no more cells could be formed, and that points still remained on the front). In such a situation, you cycle through as many attempts as necessary (with all the options for opening the "pockets" as before) to complete the mesh. Once the mesh is completed, you write out the grid files, and your gridding is done.

Some Guidelines for Constructing Patches for VGRID

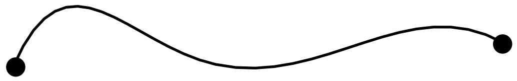
− The projected shapes of the surface patches in 3D should be as close to square, rectangle, or equilateral triangle as possible.

− The angles between patch edges should not be too wide or too acute (~ 60 < a < 120 degrees.)

− The aspect ratio of a patch should not be too high (~ < 10:1.)

− At least two of the patch edges should be approximately parallel in 4−sided patches. The patch should be defined in the .d3m file in such a way that one of the parallel edges is listed first.

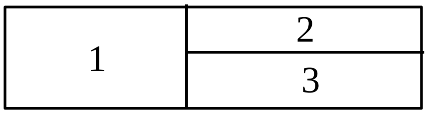
− Avoid highly (irregularly) curved edges. ==>



−Avoid irregularly shaped patches.

−Avoid edges made of three points.

−'T' intersection of patch edges is acceptable ==>



− Patches cannot have more than four edges unless they are planar.

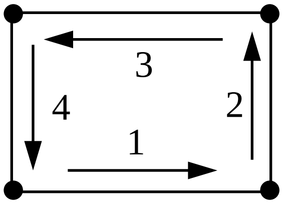
− Planar patches with any shape, edge angles, number of edges (more than 1) are acceptable.

− Patches must share edges. No duplicate edge is acceptable.

− Edges must not contain duplicate points.

− Patches must be defined with edges listed orderly and counterclockwise.

==>



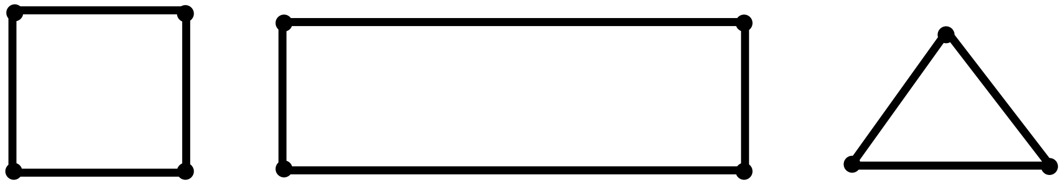
1 − 2 − 3 − 4

− The total number of patches is theoretically unlimited, but we probably don't want too many (> 1000) patches.

− The size of a patch must be larger than the local grid spacing. The patch should contain at least a few (4 or 5) triangles in each direction.

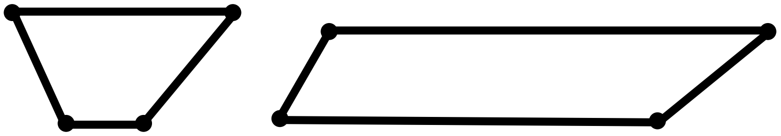
Some Examples of ‘good’ and ‘bad’ Patches for VGRID

Perfect:

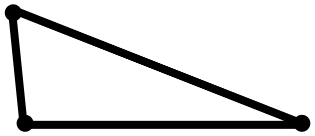


(edges do not have to be straight lines in 3D)

Good:

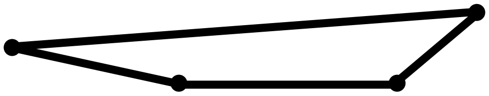


(at least 2 edges are approximately parallel)

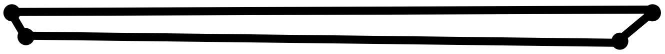


(angles not too wide/acute)

Bad:

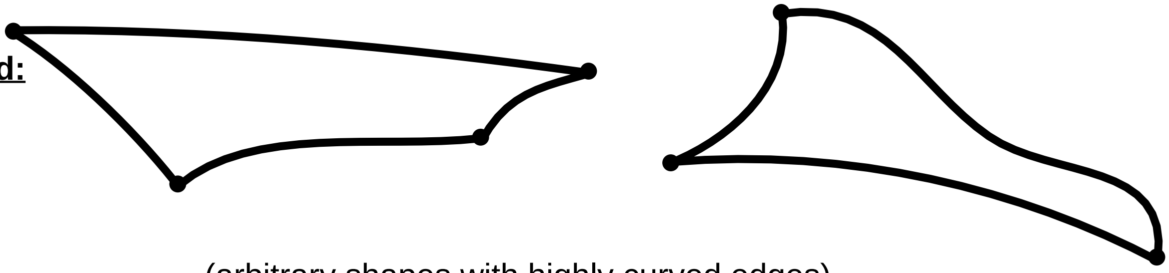


(angles are too wide/acute, none of the edges are parallel)



(aspect ratio too high)

Very bad:



(arbitrary shapes with highly curved edges)

Notes:

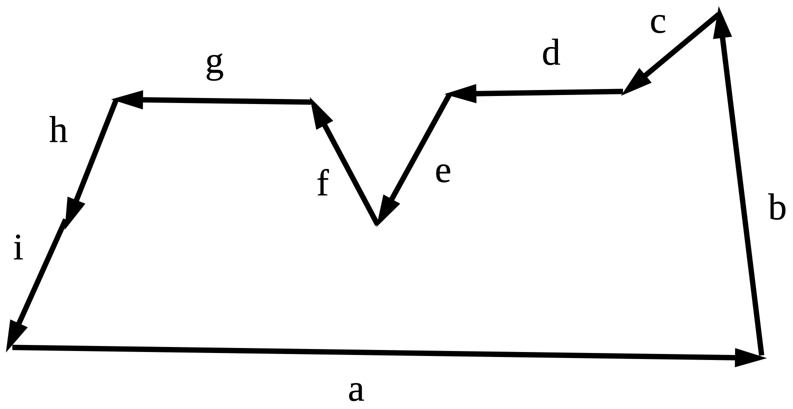
1) Assume the patches shown are all 3D and projected onto a plane.
2) For planar patches, all of the above patches are acceptable.

Additional guideline when constructing n−sided (planar) patches for VGRID:

Construct the n−sided patch such that the side defined as side number 1 forms a concave corner with the side preceding it (i.e., the side that will form the (n−1)th side of the patch).

To illustrate:

Given the following planar patch shape, with each side labeled as shown below



In order to determine which "side" of the patch to generate the surface mesh on (that is, whether to generate the surface mesh in the interior area bounded by the specified curves, or "outside" it), VGRID takes the cross product of the vector formed by the first two points of the curve chosen as side number 1, and the vector formed by the last two points of the curve chosen as the nth side. As such, for the patch shown above, the following would constitute bad choices for being marked as "side 1" :

side d : since it forms an angle > 180 degrees relative to side c

side f : since it forms an angle > 180 degrees relative to side e

side i : since it forms an angle = 180 degrees relative to side h

All other sides would be acceptable choices to be marked as "side number 1."

VGRID Patch Types:

Type	Shape
1	Planar (fewer than 3 or greater than 4 sides, OR via "explicit" assignment)
4	3−sided
5	4−sided

"Rule of Thumb" for determining a_n values for nodal sources on the Box corners:

$$3\sum_{i=1}^{A}(a_n)_i = \sum_{j=1}^{B}(a_n)_j$$

where : A = number of sources (nodal and linear) in the interior (i.e., all sources on the configuration, and NOT on the "Box")

B = number of nodal sources on the "Box" corners

Note that this rule only provides an initial guess for obtaining a reasonable balance between the strengths of the interior sources and those at the outer boundary. Usually, a few iterations (trial−and−error) are required to obtain an optimum grid distribution.

<u>INPUT AND OUTPUT FILES</u>
<u>*GridTool and VGRID*</u>
<u>*(Version 3.2)*</u>

I ***<u>GridTool</u>*** **** To set up patching and background grid (spacing) information****

Output

.d3m : *VGRID* input file
.mapbc : List of patches and BC on each

II ***<u>VGRID</u>*** **** To obtain initial (<u>unprojected</u>) front (Surface Mesh)****

Output

.bc : List of surface triangles, the node numbers comprising each, and
the surface patch on which each lies
.front : List of UNPROJECTED (x,y,z) coordinates of each node, as well as a list of the nodes
comprising each surface triangle (SEE **<u>*NOTE*</u>** BELOW)
.cogsg : FORTRAN unformatted file containing the connectivity (as in the .bc)
and grid point coordinates (of the SURFACE mesh, at this point)

III ***<u>GridTool</u>*** **** To project initial front onto surface definition****
(to obtain projected surface mesh)

Output

.front : List of PROJECTED (x,y,z) coordinates of each node, as well as a list of the nodes
comprising each surface triangle (SEE **<u>*NOTE*</u>** BELOW)
.cogsg : FORTRAN unformatted file containing the connectivity (as in the .bc)
and PROJECTED grid point coordinates (of the SURFACE mesh, at this point)

<u>*NOTE*:</u> It is a good practice to save the UNPROJECTED surface mesh (in addition to the projected surface mesh, which will obviously be used in the final volume grid generation process). In the event that VGRID encounters problems in the volume grid generation stage, and if these problems can be traced to an error that was introduced during the projection of the initial front onto the surface definition, the user can "undo" the projection on a selected patch, rather than having to regenerate and re−project the entire surface mesh. To do this, the user must save both the projected AND unprojected surface meshes. The projected surface mesh will be saved onto the .front and .cogsg files as described above. PRIOR to writing these files, however, the user should save the UNprojected .cogsg file. This file can then be read in as an "UPDATE FRONT" for use in undoing (and re−doing) the projection on selected patches on the projected surface mesh. Naming the file something indicative of its contents (like "cogsg_unp") is highly recommended, obviously.

INPUT AND OUTPUT FILES
GridTool and VGRID
(Version 3.2)

IV *VGRID* ** To generate volume mesh**
and
POSTGRID **To complete and the mesh and improve mesh quality**

Output Options:

.grd : FORMATTED file containing a list of the (x,y,z) coordinates of each node in the entire volume (surface AND field)

.int : FORMATTED file containing the connectivity of each tetrahedron in the volume. That is, it is a list of the node numbers comprising each tetrahedron in the volume

.poin1 : *(Produced only during viscous/advancing−layers applications)* FORTRAN UNFORMATTED file containing the node numbers of the points along the "direction vectors" in the advancing−layers portion of the grid. this information is used in the determination of the minimum−distance used by the turbulence model.

OR

.cogsg : FORTRAN UNFORMATTED file containing the connectivity (as in the .int) and grid point coordinates (as in the .grd) of the VOLUME mesh, at this point

.poin1 : As described above

NOTE : the .bc file is "frozen" after the surface mesh is generated (and since it contains surface triangle connectivity information−−and no actual (x,y,z) coordinate values−−it is not affected by the projection process)

Also note that at the completion of the volume grid generation process, the .front file is a two−line file containing summary information on the mesh (number of nodes, tetrahedra, surface nodes, etc.). At any intermediate point in the volume grid generation process, however, it contains the connectivity AND (x,y,z) coordinate values of all the points on the "current" front

"Event Sequence"
GridTool and VGRID
(Version 3.2)

I **GridTool :**

Read in : Geometry Definition (IGES, PLOT3D, GRIDGEN .dba, "curves," etc. files)

Goal : Set up Surface Patches and establish surface and volume mesh spacing control ("Background Grid") information

Write out : a. GridTool restart file
b. VGRID .d3m file
c. .mapbc file for USM3D

II **VGRID :**

Read in : VGRID .d3m File

Goal : Generate a surface mesh, determine the "best" orientation of 4−sided surface patches, swap face edges (if necessary).

Write out : a. .bc file
b. .cogsg file
c. .front file
d. "new" .d3m file (to save the "best" orientations of the 4−sided patches determined during the patch−by−patch interactive surface mesh generation)

NOTE: VGRID will automatically write all these files out upon completion of the surface grid generation if the user chooses to "Save the Grid and Stop."

** If the surface mesh is adequate (in terms of the number, quality, and distribution of surface triangles), the user should proceed to stage III. Otherwise, modifications to the patching and/or background grid source distribution should be performed. To do this, read the GridTool restart file into GridTool, make the changes, and write out the new .d3m (and restart) file, as in stage I. Then, return to stage II and generate a new surface mesh. This (iterative) process continues, obviously, until a surface mesh that is deemed adequate is generated. **

** Once an adequate surface mesh is generated, it is also good practice to make a copy of the .cogsg file. At this stage of the process, it contains the coordinates and connectivity of the UNPROJECTED surface mesh points. Saving a copy of this file allows the user to "undo" projections (carried out in stage III) in the event that problems occur during the volume grid generation process that can be traced to errors introduced during the projection process (see the "NOTE" in the "Input and Output Files" pages for stage III). **

"Event Sequence"
GridTool and VGRID
(Version 3.2)

III **GridTool** :

Read in : a. GridTool restart file
b. VGRID .d3m file (read in as an "update d3m")
c. front file (read in as a "VGRID Front")

Goal : To project the surface mesh onto the original surface definition (to obtain the projected surface mesh).

NOTE : Reading in the "new" .d3m file into GridTool as an "update" ensures that the patch orientations that yielded the "best" surface mesh on the 4−sided patches are reflected in the restart file.

Write out : a. GridTool restart file (to save patch orientations)
b. front file, as a "VGRID Front" (to save the projected mesh)

NOTE: GridTool will automatically re−write the .cogsg file when the .front file is written out (as a "VGRID front") after projection. A FAST−format unstructured surface mesh file is also written out automatically.

IV ***VGRID*** :

Read in : VGRID .d3m file

NOTE : VGRID will search for (and read) the corresponding .front, .cogsg, and .bc files.

Goal : Generate a volume mesh.

Write out : a. .cogsg file
b. .front file
c. .bc file
d. .poin1 file (*for viscous/advancing−layers applications)*

NOTES: VGRID will automatically write all these files out upon completion of the volume grid generation if the user chooses to "Save the Grid and Stop."

When generating an inviscid mesh, the volume grid generation process may be performed entirely in batch mode. When generating a viscous mesh, however, the generation of the viscous portion of the volume must be performed interactively (at the present point in the code’s development). Also, when generating a viscous mesh, it is recommended that the viscous portion of the volume mesh (i.e., the "advancing layers" portion) be saved (written out) before proceeding to the inviscid (advancing front) portion. Once the viscous portion is saved, the user should run VGRID again, and generate the the inviscid portion. This "second" run to obtain the inviscid portion may be performed entirely in batch.

"Event Sequence"
GridTool and VGRID
(Version 3.2)

V **POSTGRID** :

Read in : VGRID .d3m file

NOTE : POSTGRID will search for (and read) the corresponding .front, .cogsg, and .bc files.

Goal : To complete, improve and view the volume mesh.

Write out : a. .cogsg file
b. .front file
c. .bc file
d. .poin1 file (for viscous/advancing−layers applications)

OR

d. .grd file
e. .int file
f. .bc file
g. .front file
h. .poin1 file (for viscous/advancing−layers applications)

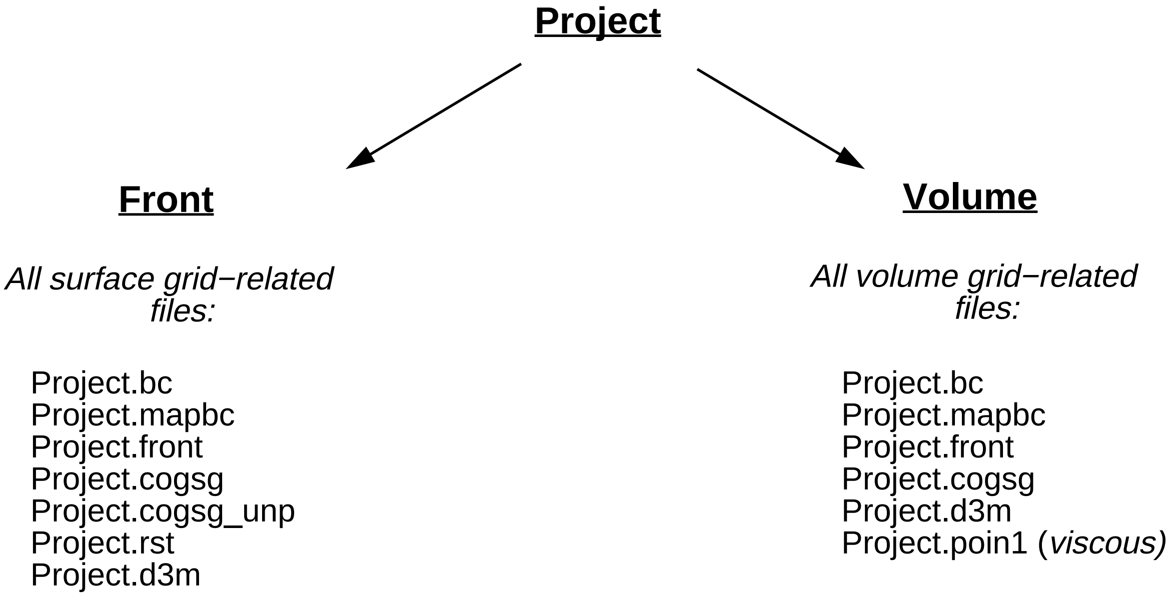
NOTE : Please refer to the "Input and Output Files" pages for more detailed descriptions of the contents of each of the above−mentioned files.

Miscellaneous Notes :

Given that the user does not know a priori how many nodes/tetrahedra will result from the volume mesh generation process, it is suggested that rather than spending the time to project the surface mesh (in stage III), the user should proceed to the volume mesh generation (stage IV) to generate a volume. This will determine if the resulting volume mesh is "too large," as well as point out any problem areas that might require re−gridding of the surface (which would mean returning to stage I).

Thus, it is good practice to save all the unprojected surface mesh files (from stage II), so that if the volume mesh derived from it turns out to be acceptable (in terms of the mesh size and quality), the user can return to stage III to perform the projection and proceed to obtain the volume as before, or use the "Moving Grid" option in POSTGRID to "update" the unprojected volume mesh with the projected coordinates contained in the projected .front file in order to automatically "move"/adjust the (viscous or inviscid) volume mesh to accommodate the projected surface mesh.

Suggested Directory Structure for Generating Grids with VGRID



Project

Front

All surface grid−related files:

Project.bc
Project.mapbc
Project.front
Project.cogsg
Project.cogsg_unp
Project.rst
Project.d3m

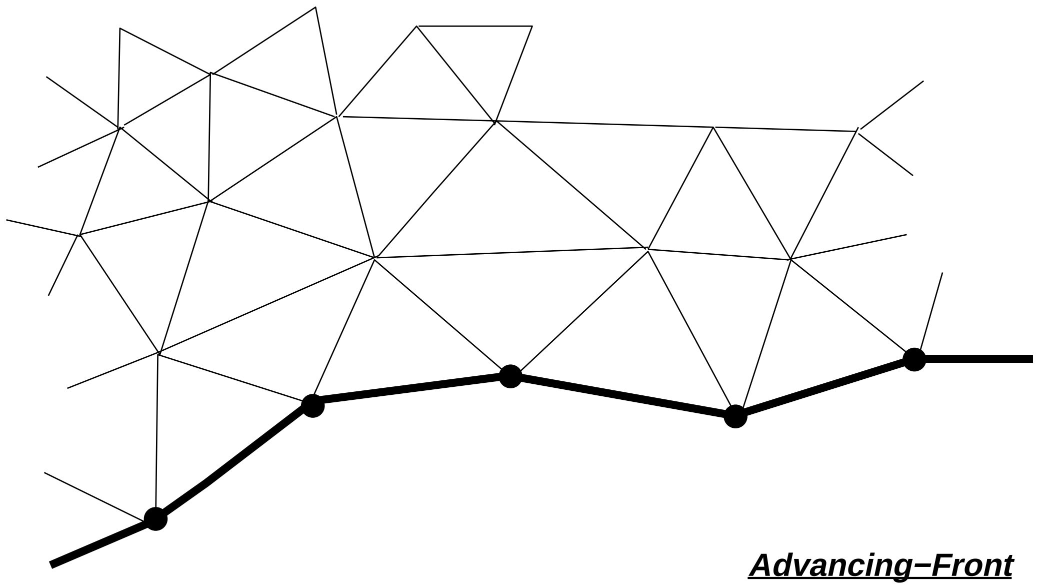
Volume

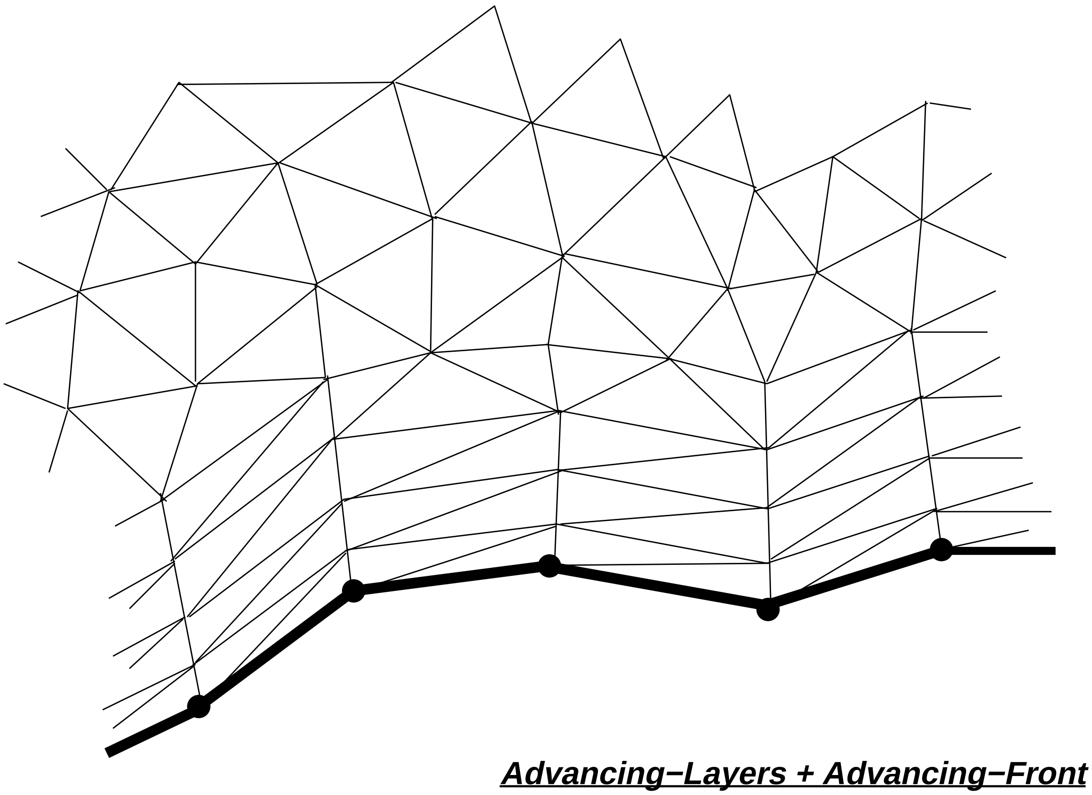
All volume grid−related files:

Project.bc
Project.mapbc
Project.front
Project.cogsg
Project.d3m
Project.poin1 (*viscous)*

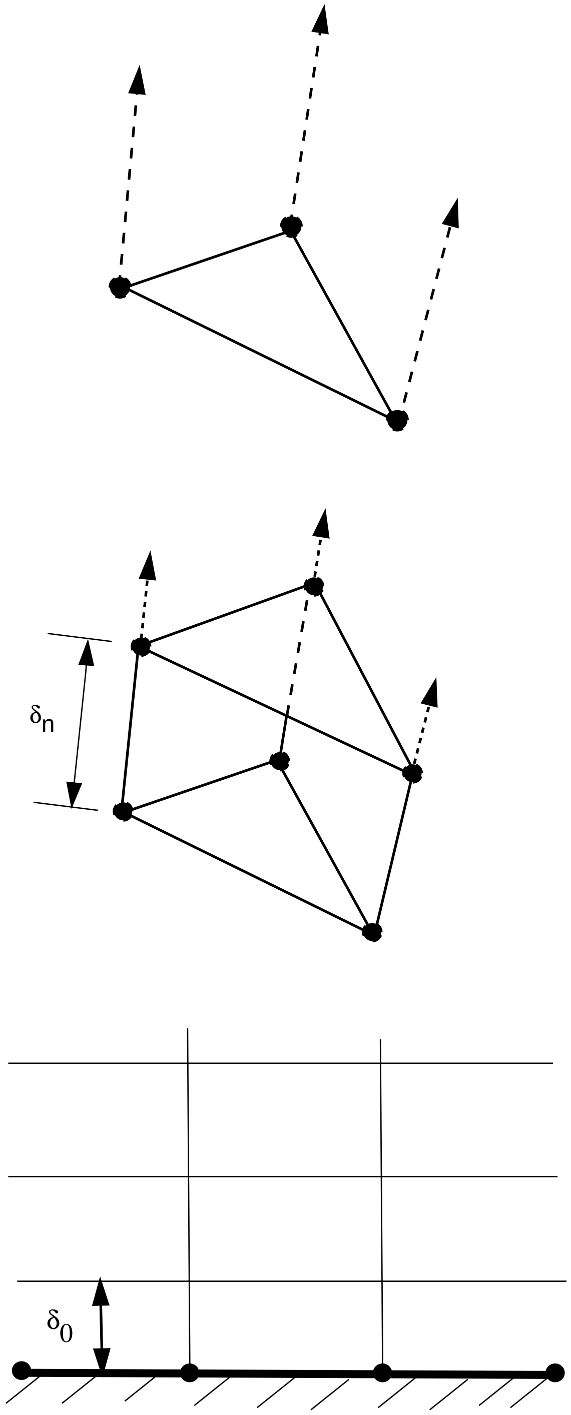
Grid Files That You Need to Proceed with the Flow Solution:

Project.mapbc
Project.bc
Project.cogsg
Project.poin1 (*viscous)*



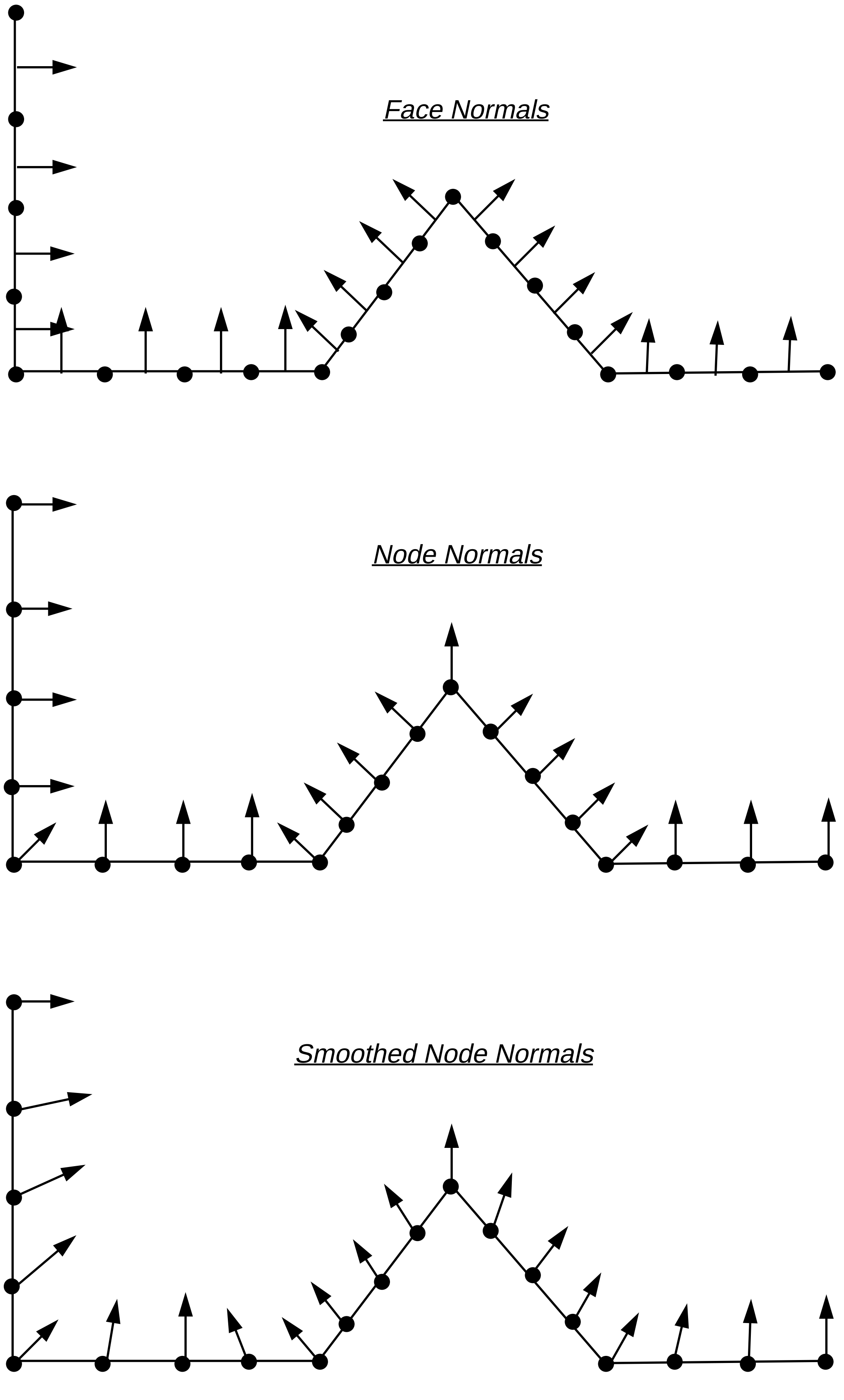


Comparison of "inviscid" (advancing−front) and "viscous" (advancing−layers + advancing−front) meshes



$$\delta_n = \delta_0 \, [\, 1 + r_1 \, (1{+}r_2)^{\,n-1} \,]^{\,n-1}$$

Distributing grid points within the advancing–layers portion of the grid

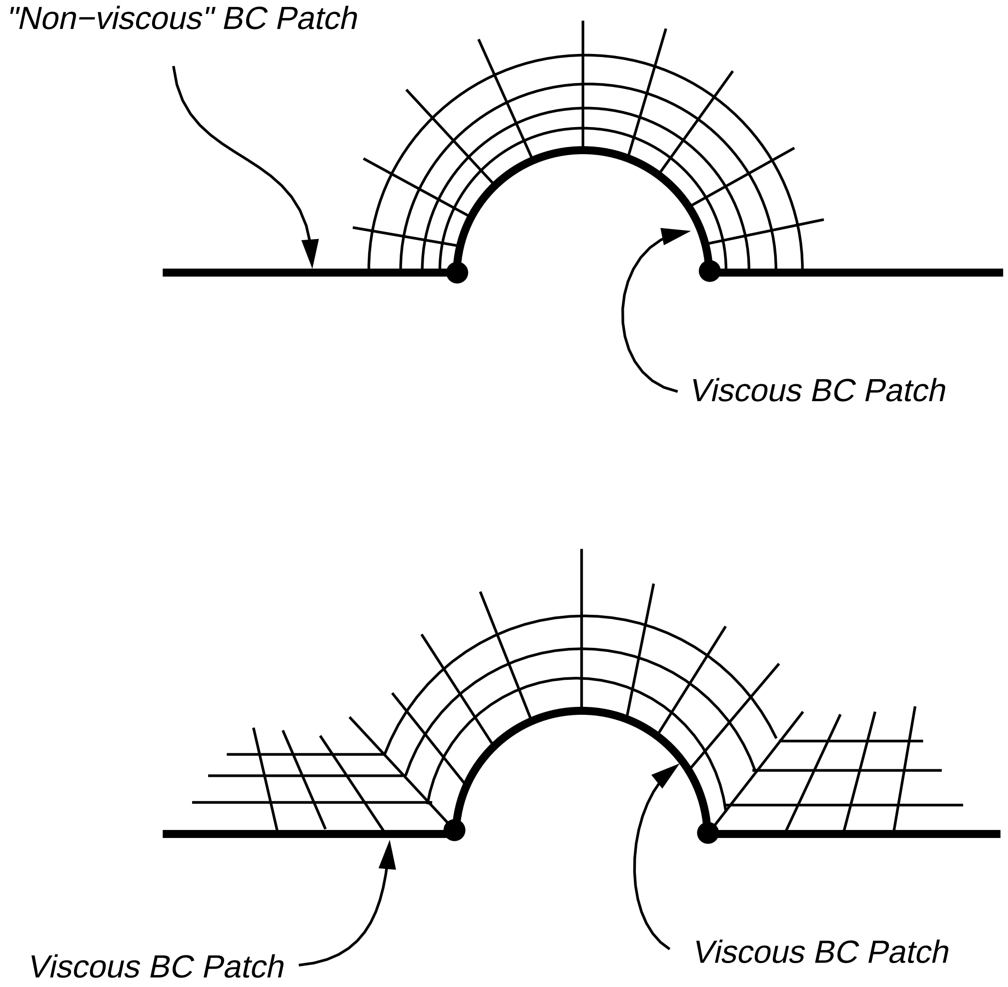


Face Normals

Node Normals

Smoothed Node Normals

Smoothing of vectors along which advancing−layers are propagated



Comparison of layer structure on adjacent patches depending on the user−specified patch boundary−condition